Improving Expert Predictions with Conformal Prediction

Eleni Straitouri [1] **Lequn Wang** [2] **Nastaran Okati** [1] **Manuel Gomez Rodriguez** [1]

Abstract

Automated decision support systems promise to help human experts solve multiclass classification tasks more efficiently and accurately. However, existing systems typically require experts to understand when to cede agency to the system or when to exercise their own agency. Otherwise, the experts may be better off solving the classification tasks on their own. In this work, we develop an automated decision support system that, by design, does not require experts to understand when to trust the system to improve performance. Rather than providing (single) label predictions and letting experts decide when to trust these predictions, our system provides sets of label predictions constructed using conformal prediction—prediction sets—and forcefully asks experts to predict labels from these sets. By using conformal prediction, our system can precisely trade-off the probability that the true label is not in the prediction set, which determines how frequently our system will mislead the experts, and the size of the prediction set, which determines the difficulty of the classification task the experts need to solve using our system. In addition, we develop an efficient and near-optimal search method to find the conformal predictor under which the experts benefit the most from using our system. Simulation experiments using synthetic and real expert predictions demonstrate that our system may help experts make more accurate predictions and is robust to the accuracy of the classifier the conformal predictor relies on.

1. Introduction

In recent years, there has been an increasing interest in developing automated decision support systems to help human experts solve tasks in a wide range of critical domains, from medicine (Jiao et al., 2020) and drug discovery (Liu et al., 2021) to candidate screening (Wang et al., 2022) and criminal justice (Grgić-Hlača et al., 2019), to name a few. Among them, one of the main focuses has been multiclass classification tasks, where a decision support system uses a classifier to make label predictions and the experts decide when to follow the predictions made by the classifier (Bansal et al., 2019; Lubars & Tan, 2019; Bordt & von Luxburg, 2020).

However, these systems typically require human experts to understand when to trust a prediction made by the classifier. Otherwise, the experts may be better off solving the classification tasks on their own (Suresh et al., 2020). This follows from the fact that, in general, the accuracy of a classifier differs across data samples (Raghu et al., 2019). In this context, several recent studies have analyzed how factors such as model confidence, model explanations and overall model calibration modulate trust (Papenmeier et al., 2019; Wang & Yin, 2021; Vodrahalli et al., 2022). Unfortunately, it is not yet clear how to make sure that the experts do not develop a misplaced trust that decreases their performance (Yin et al., 2019; Nourani et al., 2020; Zhang et al., 2020). In this work, we develop a decision support system for multiclass classification tasks that, by design, does not require experts to understand when to trust the system to improve their performance.

Our contributions. For each data sample, our decision support system provides a set of label predictions—a prediction set—and forcefully asks human experts to predict a label from this set[1]. We view this type of decision support system as more natural since, given a set of alternatives, experts tend to narrow down their options to a subset of them before making their final decision (Wright & Barbour, 1977; Beach, 1993; Ben-Akiva & Boccara, 1995). In a way, our support system helps experts by automatically narrowing down their options for them, decreasing their cognitive load and allowing them to focus their attention where it is most needed. This could be particularly useful when the task is tedious or requires domain knowledge since it is difficult to outsource the task, and domain experts are often a scarce

[1]Max Planck Institute for Software Systems, Kaiserslautern, Germany [2]Department of Computer Science, Cornell University, Ithaca, United States. Correspondence to: Eleni Straitouri <estraitouri@mpi-sws.org>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

[1]There are many systems used everyday by experts (*e.g.*, pilots flying a plane) that, under normal operation, restrict their choices. This does not mean that, in extreme circumstances, the expert should not have the ability to essentially switch off the system.

resource. In the context of clinical text annotation[2], a recent empirical study has concluded that, in terms of the overall accuracy, it may be more beneficial to recommend a subset of options than a single option (Levy et al., 2021).

By using the theory of conformal prediction (Vovk et al., 2005; Angelopoulos & Bates, 2021) to construct the above prediction set, our system can precisely control the trade-off between the probability that the true label is not in the prediction set, which determines how frequently our system will mislead an expert, and the size of the prediction set, which determines the difficulty of the classification task the expert needs to solve using our system. In this context, note that, if our system would not forcefully ask the expert to predict a label from the prediction set, it would not be able to have this level of control and good performance would depend on the expert developing a good sense on when to predict a label from the prediction set and when to predict a label from outside the set, as noted by Levy et al. (2021). In addition, given an estimator of the expert's success probability for any of the possible prediction sets, we develop an efficient and near-optimal search method to find the conformal predictor under which the expert is guaranteed to achieve the greatest accuracy with high probability. In this context, we also propose a practical method to obtain such an estimator using the confusion matrix of the expert predictions in the original classification task and a given discrete choice model.

Finally, we perform simulation experiments using synthetic and real expert predictions on several multiclass classification tasks. The results demonstrate that our decision support system is robust to both the accuracy of the classifier and the estimator of the expert's success probability it relies on—the competitive advantage it provides improves with their accuracy, and the human experts do not decrease their performance by using the system even if the classifier or the estimator are very inaccurate. Additionally, the results also show that, even if the classifiers that our system relies on have high accuracy, an expert using our system may achieve significantly higher accuracy than the classifiers on their own—in our experiments with real data, the relative reduction in misclassification probability is over 72%. Finally, by using our system, our results suggest that the expert would reduce their misclassification probability by 80%[3].

Further related work. Our work builds upon further related work on distribution-free uncertainty quantification, reliable classification and learning under algorithmic triage.

There exist three fundamental notions of distribution-free uncertainty quantification in the literature: calibration, confidence intervals, and prediction sets (Vovk et al., 2005; Balasubramanian et al., 2014; Gupta et al., 2020; Angelopoulos & Bates, 2021). Our work is most closely related to the rapidly increasing literature on prediction sets (Romano et al., 2019; 2020; Angelopoulos et al., 2021; Podkopaev & Ramdas, 2021), however, to the best of our knowledge, prediction sets have not been optimized to serve automated decision support systems such as ours. In this context, we acknowledge that Babbar et al. (2022) have also very recently proposed using prediction sets in decision support systems. However, in contrast to our work, for each data sample, they allow the expert to predict label values outside the recommended subset, *i.e.*, to predict any alternative from the entire universe of alternatives, and do not optimize the probability that the true label belongs to the subset. As a result, their method is not directly comparable to ours[4].

There is an extensive line of work on reliable or cautious classification (Del Coz et al., 2009; Liu et al., 2014; Yang et al., 2017; Mortier et al., 2021; Ma & Denoeux, 2021; Nguyen & Hüllermeier, 2021). Reliable classification aims to develop models that can provide set-valued predictions to account for the prediction uncertainty of a classifier. However, in this line of work, there are no human experts who make the final predictions given the set-valued predictions, in contrast with our work. Moreover, the set-valued predictions typically lack distribution-free guarantees.

Learning under algorithmic triage seeks the development of machine learning models that operate under different automation levels—models that make decisions for a given fraction of instances and leave the remaining ones to human experts (Raghu et al., 2019; Mozannar & Sontag, 2020; De et al., 2020; 2021; Okati et al., 2021). This line of work has predominantly focused on supervised learning settings with a few very recent notable exceptions (Straitouri et al., 2021; Meresht et al., 2022). However, in this line of work, each sample is either predicted by the model or by the human expert. In contrast, in our work, the model helps the human predict each sample. That being said, it is worth noting that there may be classifiers, data distributions and conformal scores under which the optimal conformal predictor and the optimal triage policy coincide, *i.e.*, the optimal conformal predictor does recommend a single label or the entire label set of labels.

[2]Clinical text annotation is a task where medical experts aim to identify clinical concepts in medical notes and map them to labels in a large ontology.

[3]An open-source implementation of our system is available at https://github.com/Networks-Learning/improve-expert-predictions-conformal-prediction.

[4]In our simulation experiments, we estimate the performance achieved by an expert using our system via a model-based estimator of the expert's success probability. Therefore, to compare our system with the system by Babbar et al. (2022), we would need to model the expert's success probability whenever the expert can predict any label given a prediction set, a problem for which discrete choice theory provides little guidance.

2. Problem Formulation

We consider a multiclass classification task where a human expert observes a feature vector[5] $x \in \mathcal{X}$, with $x \sim P(X)$, and needs to predict a label $y \in \mathcal{Y} = \{1, \ldots, n\}$, with $y \sim P(Y|X)$. Then, our goal is to design an automated decision support system $\mathcal{C} : \mathcal{X} \to 2^{\mathcal{Y}}$ that, given a feature vector $x \in \mathcal{X}$, helps the expert by automatically narrowing down the set of potential labels to a subset of them $\mathcal{C}(x) \subseteq \mathcal{Y}$ using a trained classifier $\hat{f} : \mathcal{X} \to [0,1]^{|\mathcal{Y}|}$ that outputs scores for each class (*e.g.*, softmax scores)[6]. The higher the score $\hat{f}_y(x)$, the more the classifier believes the true label $Y = y$. Here, we assume that, for each $x \sim P(X)$, the expert predicts a label $\hat{Y}$ among those in the subset $\mathcal{C}(x)$ according to an unknown policy $\pi(x, \mathcal{C}(x))$. More formally, $\hat{Y} \sim \pi(x, \mathcal{C}(x))$, where $\pi : \mathcal{X} \times 2^{\mathcal{Y}} \to \Delta(\mathcal{Y})$ and $\Delta(\mathcal{Y})$ denotes the probability simplex over the set of labels $\mathcal{Y}$, and $\pi_y(x, \mathcal{C}(x)) = 0$ if $y \notin \mathcal{C}(x)$. Refer to Figure 1 for an illustration of the automated decision support system we consider.

Ideally, we would like that, by design, the expert can only benefit from using the automated decision support system $\mathcal{C}$, *i.e.*,

$$\mathsf{P}[\hat{Y} = Y \,;\, \mathcal{C}] \leq \mathsf{P}[\hat{Y} = Y \,;\, \mathcal{Y}], \tag{1}$$

where $\mathsf{P}[\hat{Y} = Y \,;\, \mathcal{C}]$ denotes the expert's success probability if, for each $x \sim P(X)$, the human expert predicts a label $\hat{Y}$ among those in the subset $\mathcal{C}(x)$. However, not all automated decision support systems fulfilling the above requirement will be equally useful—some will help experts increase their success probability more than others. For example, a system that always recommends $\mathcal{C}(x) = \mathcal{Y}$ for all $x \in \mathcal{X}$ satisfies Eq. 1. However, it is useless to the experts. Therefore, among those systems satisfying Eq. 1, we would like to find the system $\mathcal{C}$ that helps the experts achieve the highest success probability[7], *i.e.*,

$$\mathcal{C} = \operatorname*{argmax}_{\mathcal{C}} \mathsf{P}[\hat{Y} = Y \,;\, \mathcal{C}]. \tag{2}$$

To address the design of such a system, we will look at the problem from the perspective of conformal prediction (Vovk et al., 2005; Angelopoulos & Bates, 2021).

3. Subset Selection using Conformal Prediction

In general, if the trained classifier $\hat{f}$ we use to build $\mathcal{C}(X)$ is not perfect, the true label Y may or may not be included in $\mathcal{C}(X)$. In what follows, we will construct the subsets $\mathcal{C}(X)$ using the theory of conformal prediction. This will allow our system to be robust to the accuracy of the classifier $\hat{f}$ it uses—the probability $\mathsf{P}[Y \in \mathcal{C}(X)]$ that the true label Y belongs to the subset $\mathcal{C}(X) = \mathcal{C}_\alpha(X)$ will match almost exactly a given target probability $1 - \alpha$ with high probability, without making any distributional assumptions about the data distribution $P(X)P(Y|X)$ nor the classifier $\hat{f}$.

Let $\mathcal{D}_{\text{cal}} = \{(x_i, y_i)\}_{i=1}^m$ be a calibration set, where $(x_i, y_i) \sim P(X)P(Y|X)$, $s(x_i, y_i) = 1 - \hat{f}_{y_i}(x_i)$ be the *conformal score*[8] (*i.e.*, if the classifier is catastrophically wrong, the conformal score will be close to one), and $\hat{q}$ be the $\frac{\lceil (m+1)(1-\alpha) \rceil}{m}$ empirical quantile of the conformal scores $s(x_1, y_1), \ldots, s(x_m, y_m)$. Then, if we construct the subsets $\mathcal{C}_\alpha(X)$ for new data samples as follows:

$$\mathcal{C}_\alpha(X) = \{y | s(X, y) \leq \hat{q}\}, \tag{3}$$

we have that the probability that the true label Y belongs to the subset $\mathcal{C}_\alpha(X)$ conditionally on the calibration set $\mathcal{D}_{\text{cal}}$ is almost exactly $1 - \alpha$ with high probability as long as the size m of the calibration set is sufficiently large. Specifically, we first note that the coverage probability is a random quantity[9] whose distribution is given by the following proposition (refer to Appendix A.5 in Hulsman (2022) for the proof):

Proposition 3.1. *For a decision support system $\mathcal{C}$ that constructs the subsets $\mathcal{C}_\alpha(X)$ using Eq. 3, it holds that*

$$\mathsf{P}[Y \in \mathcal{C}_\alpha(X) | \mathcal{D}_{cal}] \sim Beta(\lceil (m+1)(1-\alpha) \rceil, \lfloor (m+1)\alpha \rfloor) \tag{4}$$

as long as the conformal scores $s(X_i, Y_i)$ for all $(X_i, Y_i) \in \mathcal{D}_{cal}$ are almost surely distinct.

As an immediate consequence of Proposition 3.1, using the definition of the beta distribution, we have that

$$1 - \alpha \leq \mathsf{E}[\mathsf{P}[Y \in \mathcal{C}_\alpha(X) | \mathcal{D}_{\text{cal}}]] = 1 - \frac{\lfloor (m+1)\alpha \rfloor}{m+1} \leq 1 - \alpha + \frac{1}{m+1}.$$

Moreover, given a target probability $1 - \alpha$ and tolerance values $\delta, \epsilon \in (0, 1)$, we can compute the minimum size m of the calibration set $\mathcal{D}_{\text{cal}}$ such that $\mathcal{C}$ enjoys Probably Approximately Correct (PAC) coverage guarantees, *i.e.*, with probability $1 - \delta$, it holds that (Angelopoulos & Bates, 2021)

$$1 - \alpha - \epsilon \leq \mathsf{P}[Y \in \mathcal{C}_\alpha(X) | \mathcal{D}_{\text{cal}}] \leq 1 - \alpha + \epsilon.$$

While the above coverage guarantee is valid for any choice of α value, we would like to emphasize that there may be

[5] We denote random variables with capital letters and realizations of random variables with lower case letters.

[6] The assumption that $\hat{f}(x) \in [0,1]^n$ is without loss of generality.

[7] Note that maximizing the expert's success probability $\mathsf{P}[\hat{Y} = Y \,;\, \mathcal{C}]$ is equivalent to minimizing the expected 0-1 loss $\mathsf{E}[\mathsf{I}(\hat{Y} \neq Y) \,;\, \mathcal{C}]$. Considering other types of losses is left as an interesting avenue for future work.

[8] In general, the conformal score $s(x, y)$ can be any function of x and y measuring the *similarity* between samples.

[9] The randomness comes from the randomness of the calibration set sampling.

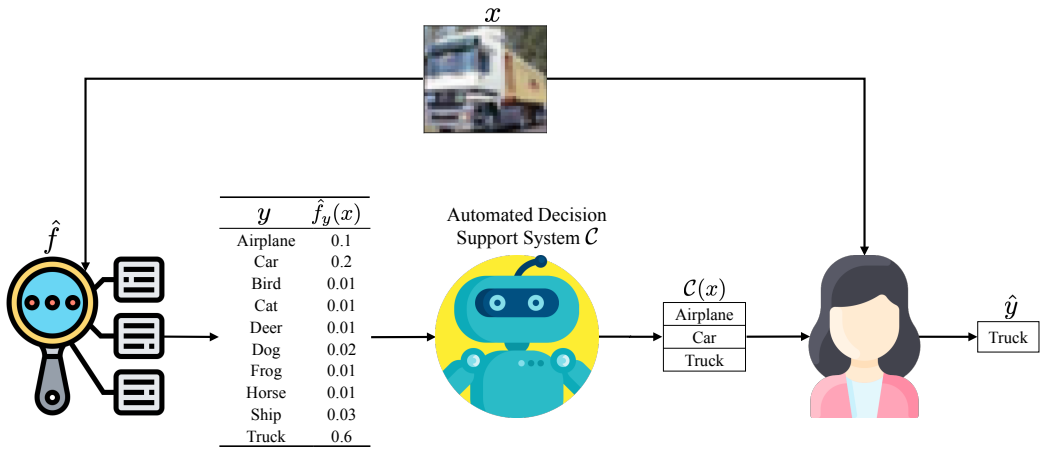


Figure 1. Our automated decision support system $\mathcal{C}$. Given a sample with a feature vector x, our system $\mathcal{C}$ narrows down the set of potential labels $y \in Y$ to a subset of them $\mathcal{C}(x)$ using the scores $\hat{f}_y(x)$ provided by a classifier $\hat{f}$ for each class y. The human expert receives the recommended subset $\mathcal{C}(x)$, together with the sample, and predicts a label $\hat{y}$ from $\mathcal{C}(x)$ according to a policy $\pi(x, \mathcal{C}(x))$.

some α values that will lead to larger gains in terms of success probability $\mathrm{P}[\hat{Y} = Y; \mathcal{C}]$ than others. Therefore, in what follows, our goal is to find the optimal α that maximizes the expert's success probability given a calibration set D_{cal}.

Remark. Most of the literature on conformal prediction focuses on the following conformal calibration guarantee (refer to Appendix D in Angelopoulos & Bates (2021) for the proof):

Theorem 3.2. *For an automated decision support system $\mathcal{C}$ that constructs the subsets $\mathcal{C}(X)$ using Eq. 3, it holds that*

$$1 - \alpha \leq \mathrm{P}[Y \in \mathcal{C}(X)] \leq 1 - \alpha + \frac{1}{m+1},$$

where the probability is over the randomness in the sample it helps predicting and the calibration set used to compute the empirical quantile $\hat{q}$.

However, to afford the above marginal guarantee in our work, we would be unable to optimize the α value to maximize the expert' s success probability given a calibration set D_{cal}. This is because the guarantee requires that α and D_{cal} are independent. That being said, in our experiments, we have empirically found that the optimal α does not vary significantly across calibration sets, as shown in Appendix E.

4. Optimizing Across Conformal Predictors

We start by realizing that, given a calibration set $D_{\text{cal}} = \{(x_i, y_i)\}_{i=1}^{m}$, there only exist m different conformal predictors. This is because the empirical quantile $\hat{q}$, which the subsets $\mathcal{C}(x_i)$ depend on, can only take m different values. As a result, to find the optimal conformal predictor that maximizes the expert's success probability, we need to solve the following maximization problem:

$$\alpha^* = \operatorname*{argmax}_{\alpha \in A} \mathrm{P}[\hat{Y} = Y; \mathcal{C}], \tag{5}$$

where $A = \{\alpha_i\}_{i \in [m]}$, with $\alpha_i = 1 - i/(m+1)$, and the probability is only over the randomness in the samples the system helps predicting.

However, to find a near optimal solution $\hat{\alpha}$ to the above problem, we need to estimate the expert's success probability $\mathrm{P}[\hat{Y} = Y; \mathcal{C}]$. Assume for now that, for each $\alpha \in A$, we have access to an estimator $\hat{p}$ of the expert's success probability such that, for any $\delta \in (0, 1)$, with probability at least $1 - \delta$, it holds that $|\hat{p} - \mathrm{P}[\hat{Y} = Y; \mathcal{C}]| \leq \epsilon_{\alpha, m}$. Then, we can use the following proposition to find a near-optimal solution $\hat{\alpha}$ to Eq. 5 with high probability:

Proposition 4.1. *For any $\delta \in (0, 1)$, consider an automated decision support system $\mathcal{C}_\wedge$ with*

$$\hat{\alpha} = \operatorname*{argmax}_{\alpha \in A} \hat{p} - \epsilon_{\alpha, m}. \tag{6}$$

With probability at least $1 - \delta$, it holds that $\mathrm{P}[\hat{Y} = Y; \mathcal{C}_\wedge] \geq \mathrm{P}[\hat{Y} = Y; \mathcal{C}] - 2\epsilon_{\alpha, m}$ $\forall \alpha \in A$ simultaneously.

More specifically, the above result directly implies that for any $\delta \in (0, 1)$, with probability at least $1 - \delta$, it holds that:

$$\mathrm{P}[\hat{Y} = Y; \mathcal{C}^*] - \mathrm{P}[\hat{Y} = Y; \mathcal{C}_\wedge] \leq 2\epsilon_{\alpha^*, m}. \tag{7}$$

Here, note that the above guarantees do not make use of the PAC coverage guarantees afforded by conformal prediction—they hold for any parameterized set-value predictor.

Algorithm 1 Finding a near-optimal $\hat{\alpha}$

Input: $\hat{f}$, D_{est}, D_{cal}, δ, m
Initialize: $A = \{\}$, $\hat{\alpha}$ 0, t 0
for i 1, .., m **do**
  α 1 $\frac{i}{m+1}$
  A $A \cup \{\alpha\}$
end for
for α A **do**
  $\hat{p}$, ϵ ; $=m$ ESTIMATE($\alpha, \delta, D_{\text{est}}, D_{\text{cal}}, \hat{f}$)
  if t $\hat{p}$ ϵ ; $=m$ **then**
    t $\hat{p}$ ϵ ; $=m$
    $\hat{\alpha}$ α
  end if
end for
return $\hat{\alpha}$

In what follows, we propose a practical method to estimate the expert's success probability $\mathrm{P}[\hat{Y} = Y ; C]$ that builds upon the multinomial logit model (MNL), one of the most popular models in the vast literature on discrete choice models (Heiss, 2016). More specifically, given a sample (x, y), we assume that the expert's conditional success probability for the subset $C(x)$ is given by

$$\mathrm{P}[\hat{Y} = y ; C \mid y \in C(x)] = \frac{e^{u_{yy}}}{\sum_{y' \in C(x)} e^{u_{yy'}}}, \quad (8)$$

where $u_{yy'}$ denotes the expert's preference for label value $y' \in Y$ whenever the true label is y. In the language of discrete choice models, one can view the true label y as the *context* in which the expert chooses among alternatives (Tversky & Simonson, 1993). In Appendix I, we consider and experiment with a more expressive context that, in addition to the true label, distinguishes between different levels of *difficulty* across data samples.

Further, to estimate the parameters $u_{yy'}$, we assume we have access to (an estimation of) the confusion matrix $\boldsymbol{C}$ for the expert predictions in the (original) multiclass classification task, similarly as in Kerrigan et al. (2021), *i.e.*,

$$\boldsymbol{C} = [C_{yy'}]_{y,y' \in Y}, \text{ where } C_{yy'} = \mathrm{P}[\hat{Y} = y' ; Y \mid Y = y],$$

and naturally set $u_{yy'} = \log C_{yy'}$. Then, we can compute a Monte-Carlo estimator $\hat{p}$ of the expert's success probability $\mathrm{P}[\hat{Y} = Y ; C]$ using the above conditional success probability $\mathrm{P}[\hat{Y} = y ; C \mid y \in C(x)]$ and an estimation set $D_{\text{est}} = \{(x_i, y_i)\}_{i \in [m]}$[10], *i.e.*,

$$\hat{p} = \frac{1}{m} \sum_{i \in [m] \mid y_i \in C(x_i)} \mathrm{P}[\hat{Y} = y_i ; C \mid y_i \in C(x_i)]. \quad (9)$$

[10]The number of samples in D_{cal} and D_{est} can differ. For simplicity, we assume both sets contain m samples.

Finally, for each $\alpha \in A$, using Hoeffding's inequality[11,12], we can conclude that, with probability at least $1 - \delta$, it holds that (refer to Appendix A.2):

$$|\hat{p} - \mathrm{P}[\hat{Y} = Y ; C]| \sqrt{\frac{\log \frac{1}{\delta}}{2m}} := \epsilon_{\cdot}. \quad (10)$$

As a consequence, as $m \to \infty$, $\epsilon_{\cdot}$ converges to zero. This directly implies that the near-optimal $\hat{\alpha}$ converges to the true optimal α and that, with probability at least $1 - \delta$, our system C satisfies Eq. 1 asymptotically with respect to the number of samples m in the estimation set.

Algorithm 1 summarizes the overall search method, where the function ESTIMATE( ) uses Eqs. 9 and 10. The algorithm first builds A and then finds the near-optimal $\hat{\alpha}$ in A. To build A, it needs $O(m)$ steps. To find the near-optimal $\hat{\alpha}$, for each value $\alpha \in A$ and each sample $(x_i, y_i) \in D_{\text{est}}$, it needs to compute a subset $C(x)$. This is achieved by sorting the conformal scores and reusing computations across α values, which takes $O(m \log m + mn \log n)$ steps. Therefore, the overall time complexity is $O(m \log m + mn \log n)$.

Remarks. By using the MNL, we implicitly assume the independence of irrelevant alternatives (IIA) (Luce, 1959), an axiom that states that the expert's relative preference between two alternatives remains the same over all possible subsets containing these alternatives. While IIA is one of the most widely used axioms in the literature on discrete choice models, there is also a large body of experimental literature claiming to document real-world settings where IIA fails to hold (Tversky, 1972; Huber et al., 1982; Simonson, 1989). Fortunately, we have empirically found that experts may benefit from using our system even under strong violations of the IIA assumption in the estimator of the expert's success probability (*i.e.*, when the estimator of the expert's success probability is not accurate), as shown in Figures 3 and 4.

Conformal prediction is one of many possible ways to construct set-valued predictors (Chzhen et al., 2021), *i.e.*, predictors that, for each sample $x \in X$, output a set of label candidates $C(x)$. In our work, we favor conformal predictors over alternatives because they provably output *trustworthy* sets $C(x)$ without making any assumption about the data distribution nor the classifier they rely upon. In fact, we can use conformal predictors with any off-the-shelf classifier. However, we would like to emphasize that our efficient

[11]By using Hoeffding's inequality, we derive a fairly conservative constant error bound for all α values, however, we have experimentally found that, even with a relatively small amount of estimation and calibration data, our algorithm identifies near-optimal $\hat{\alpha}$ values, as shown in Figure 2. That being said, one could use tighter concentration inequalities such as Hoeffding–Bentkus and Waudby-Smith–Ramdas (Bates et al., 2021).

[12]We are applying Hoeffding's inequality only on the randomness of the samples (X_i, Y_i), which are independent and identically distributed.

Table 1. Empirical success probability achieved by four different experts using our system during test, each with a different success probability $\mathsf{P}[\hat{Y} = Y \mid Y]$, on four prediction tasks where the classifier achieves a different success probability $\mathsf{P}[Y^0 = Y]$. Each column corresponds to a prediction task, and each row to an expert. In each task, the number of label values $n = 10$ and the size of the calibration and estimation sets is $m = 1{,}200$. Each cell shows only the average since all standard errors are below 10^{-2}.

$\mathsf{P}[\hat{Y} = Y \mid Y]$	$\mathsf{P}[Y^0 = Y]$ 0.3	0.5	0.7	0.9
0.3	0.41	0.58	0.75	0.91
0.5	0.55	0.68	0.80	0.93
0.7	0.72	0.79	0.87	0.95
0.9	0.90	0.91	0.95	0.98

search method (Algorithm 1) is rather generic and, together with an estimator of the expert's success probability with provable guarantees, may be used to find a near-optimal set-valued predictor within a discrete set of set-valued predictors that maximizes the expert's success probability. This is because our near-optimal guarantees in Proposition 4.1 do not make use of the PAC guarantees afforded by conformal prediction, as discussed previously. In Appendix D, we discuss an alternative set-valued predictor with PAC coverage guarantees, which may provide improved performance in scenarios where the classifier underpinning our system has not particularly high average accuracy. We hope our work will encourage others to develop set-valued predictors specifically designed to serve decision support systems.

5. Experiments on Synthetic Data

In this section, we evaluate our system against the accuracy of the expert and the classifier, the size of the calibration and estimation sets, as well as the number of label values. Moreover, we analyze the robustness of our system to violations of the IIA assumption in the estimator of the expert's success probability[13].

Experimental setup. We create a variety of synthetic prediction tasks, each with 20 features per sample and a varying number of label values n and difficulty. Refer to Appendix B for more details about the prediction tasks. For each prediction task, we generate 10,000 samples, pick 20% of these samples at random as test set, which we use to estimate the performance of our system, and also randomly split the remaining 80% into three disjoint subsets for training, calibration, and estimation, whose sizes we vary across experiments. In each experiment, we specify the number of samples in the calibration and estimation sets—the remaining samples are used for training.

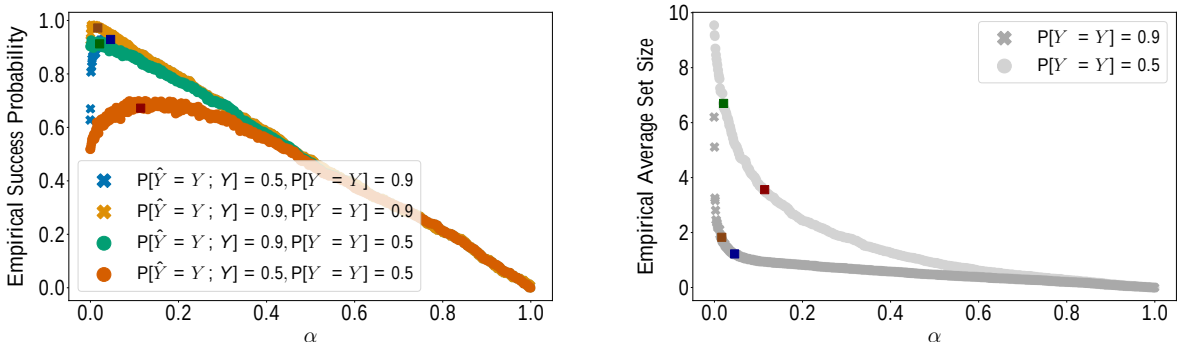
For each prediction task, we train a logistic regression model $P(Y^0 \mid X)$, which depending on the difficulty of the prediction task, achieves different success probability values $\mathsf{P}[Y^0 = Y]$. Moreover, we sample the expert's predictions $\hat{Y}$ from the multinomial logit model defined by Eq. 8, with $C_{yy} = \frac{}{n} \quad \gamma\epsilon_C$ and $C_{yy'} = \frac{1 \quad C_{yy}}{n} \quad \beta$, where π is a parameter that controls the expert's success probability $\mathsf{P}[\hat{Y} = Y \mid Y]$, $\epsilon_C \quad \mathrm{U}(0, \min(1 \quad \frac{}{n}, \frac{}{n}))$, $\beta \quad \mathrm{N}(0, ((1 \quad C_{yy})/(6n))^2)$ for all $y \neq y^0$, and γ is a normalization term. Finally, we repeat each experiment ten times and, each time, we sample different train, estimation, calibration, and test sets following the above procedure.

Experts always benefit from our system even if the classifier has low accuracy. We estimate the success probability $\mathsf{P}[\hat{Y} = Y \mid C_\wedge]$ achieved by four different experts, each with a different success probability $\mathsf{P}[\hat{Y} = Y \mid Y]$, on four prediction tasks where the classifier achieves a different success probability $\mathsf{P}[Y^0 = Y]$. Table 1 summarizes the results, where each column corresponds to a different prediction task and each row corresponds to a different expert. We find that, using our system, the expert solves the prediction task significantly more accurately than the expert or the classifier on their own. Moreover, it is rather remarkable that, even if the classifier has low accuracy, the expert always benefits from using our system—in other words, our system is robust to the performance of the classifier it relies on. In Appendix C, we show qualitatively similar results for prediction tasks with other values of n and m.

The performance of our system under $\hat{\alpha}$ found by Algorithm 1 and under α is very similar. Given three prediction tasks where the expert and the classifier achieve different success probabilities $\mathsf{P}[\hat{Y} = Y \mid Y]$ and $\mathsf{P}[Y^0 = Y]$, we compare the performance of our system under the near-optimal $\hat{\alpha}$ found by Algorithm 1 and under all other possible $\alpha \in A$ values. Figure 2 summarizes the results, which suggest that: (i) the performance under $\hat{\alpha}$ is very close to that under α , as suggested by Proposition 4.1; and, (ii) as long as $\alpha \quad \alpha$, the performance of our system increases monotonically with respect to α, however, once $\alpha > \alpha$, the performance deteriorates as we increase α. (iii) the higher the expert's success probability $\mathsf{P}[\hat{Y} = Y \mid Y]$, the smaller the near optimal $\hat{\alpha}$ and thus the greater the average size of the subsets $C_\wedge(X)$. In Appendix G, we also show that, the smaller the near optimal $\hat{\alpha}$, the greater the spread of the empirical distribution of the size of the subsets $C_\wedge(X)$. We found qualitatively similar results using other expert-classifier pairs with different success probabilities.

Our system needs a relatively small amount of calibration and estimation data. We vary the amount of calibration and estimation data m we feed into Algorithm 1

[13] All algorithms ran on a Debian machine equipped with Intel Xeon E5-2667 v4 @ 3.2 GHz, 32GB memory and two M40 Nvidia Tesla GPU cards. See Appendix B for further details.



(a) Empirical success probability vs α

(b) Empirical average set size vs α

Figure 2. Empirical success probability achieved by two different experts using our system, each with a different success probability $\mathsf{P}[\hat{Y} = Y ; Y]$, and average size of the recommended sets during test for each α 2 A on two synthetic prediction tasks where the classifier achieves a different success probability $\mathsf{P}[Y^0 = Y]$. Here, note that the empirical average set size only depends on the classifier's success probability $\mathsf{P}[Y^0 = Y]$, not the expert, and thus we only need two lines. In all experiments, the number of label values $n = 10$ and the size of the calibration and estimation sets is $m = 1{,}200$. Each marker corresponds to a different α value, and the darker points correspond to $\hat{\alpha}$. The coloring of the darker points for each prediction task is the same in both panels.

and, each time, estimate the expert's success probability $\mathsf{P}[\hat{Y} = Y ; C_\wedge]$. Across prediction tasks, we consistently find that our system needs a relatively small amount of calibration and estimation data to perform well. For example, for all prediction tasks with $n = 10$ label values and varying level of difficulty, the relative gain in empirical success probability achieved by an expert using our system with respect to an expert on their own, averaged across experts with $\mathsf{P}[\hat{Y} = Y ; Y]$ 2 $f0.3, 0.5, 0.7, 0.9g$, goes from 47.56 4.51% for $m = 160$ to 48.66 4.54% for $m = 1{,}200$.

The greater the number of label values, the more an expert benefits from using our system. We consider prediction tasks with a varying number of label values, from $n = 10$ to $n = 100$, and estimate the expert's success probability $\mathsf{P}[\hat{Y} = Y ; C_\wedge]$ for each task. Our results suggest that the relative gain in success probability, averaged across experts with $\mathsf{P}[\hat{Y} = Y ; Y]$ 2 $f0.3, 0.5, 0.7, 0.9g$, increases with the number of label values. For example, for $m = 400$, it goes from 48.36 4.50% for $n = 10$ to 69.44 5.20% for $n = 100$. For other m values, we found a similar trend.

Our system is robust to strong violations of the IIA assumption in the estimator of the expert's success probability. To study the robustness of our system to violations of the IIA assumption in the estimator of the expert's success probability, we allow the expert's preference u_{yy^0} for each label value y^0 6= y in Eq. 8 to depend on the corresponding prediction set $C_\wedge(x)$ at test time. More specifically, we set

$$u_{yy^0} = \log\left(C_{yy^0} + p\,\frac{\mathsf{I}[y^0 \neq y]}{jC_\wedge(x)nfygj}\sum_{y'' 2 C_\wedge(x)} C_{yy''}\right),$$

where p 2 $[0, 1]$ is a parameter that controls the severity of the violation of the IIA assumption at test time. Here, note that if $p = 1$, the expert does not benefit from using our system as long as the prediction set $C(x)$ 6= fyg, *i.e.*, the expert's conditional success probability is given by $\mathsf{P}[\hat{Y} = y ; C\ jy\ 2\ C(x)] = \mathsf{P}[\hat{Y} = y ; Y]$. Figure 3 summarizes the results, which show that our system is robust to (strong) violations of the IIA assumption in the estimator of the expert's success probability.

6. Experiments on Real Data

In this section, we experiment with a dataset with real expert predictions on a multiclass classification task over natural images and several popular and highly accurate deep neural network classifiers. In doing so, we benchmark the performance of our system against a competitive top-k set-valued predictor baseline, which always returns the k label values with the highest scores, and analyze its robustness to violations of the IIA assumption in the estimator of the expert's success probability. Here, we would like to explicitly note that we rely on the confusion matrix estimated using real expert predictions on the (original) multiclass classification task and the multinomial logit model defined by Eq. 8 to estimate the performance of our system and the competitive top-k set-valued predictor baseline—no real experts actually used our decision support system.

Data description. We experiment with the dataset CIFAR-10H (Peterson et al., 2019), which contains 10,000 natural images taken from the test set of the standard CIFAR-10 (Krizhevsky et al., 2009). Each of these images belongs to one of $n = 10$ classes and contains approximately 50

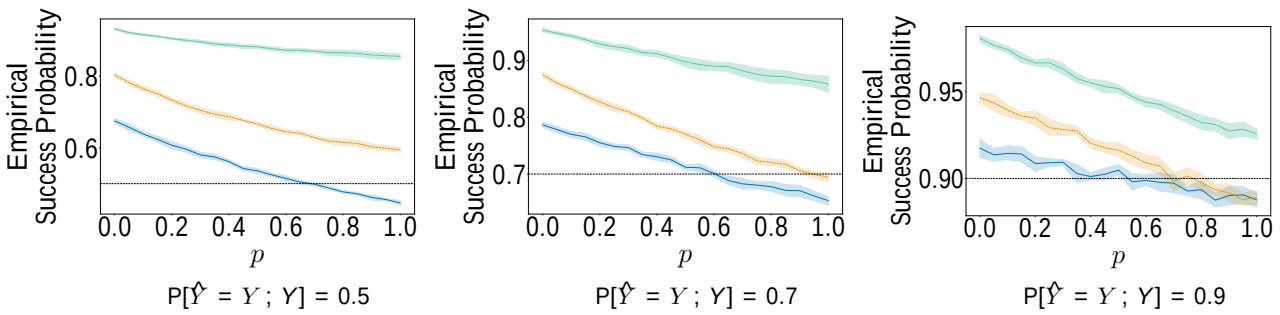


Figure 3. Empirical success probability achieved by three different experts using our system during test, each with a different success probability $\mathsf{P}[\hat{Y} = Y ; Y]$, against severity p of the violation of the IIA assumption on three prediction tasks where the classifier achieves a different success probability $\mathsf{P}[Y^0 = Y]$. In each panel, the horizontal line shows the empirical success probability achieved by the expert at solving the (original) multiclass task during test. The number of labels is $n = 10$ and the size of the calibration and estimation sets is $m = 1{,}200$. Shaded regions correspond to 95% confidence intervals.

expert predictions $\hat{Y}$[14]. Here, we randomly split the dataset into three disjoint subsets for calibration, estimation and test, whose sizes we vary across experiments. In each experiment, we use the test set to estimate the performance of our system and we specify the number of samples in the calibration and estimation sets—the remaining samples are used for testing.

Experimental setup. Rather than training a classifier, we use three popular and highly accurate deep neural network classifiers trained on CIFAR-10, namely ResNet-110 (He et al., 2016a), PreResNet-110 (He et al., 2016b) and DenseNet (Huang et al., 2017). Moreover, we use the human predictions $\hat{Y}$ to estimate the confusion matrix $\mathbf{C}$ for the expert predictions in the (original) multiclass classification task (Kerrigan et al., 2021) and then sample the expert's prediction $\hat{Y}$ from the multinomial logit model defined by Eq. 8 to both estimate the expert's conditional success probabilities in Eq. 9 in Algorithm 1 and estimate the expert's success probability during testing. In what follows, even though the expert's performance during testing is estimated using the multinomial logit model, rather than using real predictions from experts using our system, we refer to (the performance of) such a *simulated* expert as an expert.

Performance evaluation. We start by estimating the success probability $\mathsf{P}[\hat{Y} = Y ; C_{\hat{\alpha}}]$ achieved by an expert using our system ($C_{\hat{\alpha}}$) and the best top-k set-valued predictor (C_k), which returns the k label values with the highest scores[15].

[14]The dataset CIFAR-10H is among the only publicly available datasets (released under Creative Commons BY-NC-SA 4.0 license) that we found containing multiple expert predictions per sample, necessary to estimate $\mathbf{C}$, a relatively large number of samples, and more than two classes. However, since our methodology is rather general, our system may be useful in other applications.

[15]Appendix F shows the success probability achieved by an expert using the top-k set-valued predictor for different k values both for synthetic and real data.

Table 2. Empirical success probabilities achieved by three popular deep neural network classifiers and by an expert using our system ($C_{\hat{\alpha}}$) and the best top-k set-valued predictor (C_k) with these classifiers during test on the CIFAR-10H dataset. The size of the calibration and estimation sets is $m = 1{,}500$ and the expert's empirical success probability at solving the (original) multiclass task is $\mathsf{P}[\hat{Y} = Y ; Y]$ 0.947. Each cell shows only the average since the standard errors are all below 10 ^{2}.

	CLASSIFIER	$C_{\hat{\alpha}}$	C_k
RESNET-110	0.928	0.987	0.967
PRERESNET-110	0.944	0.989	0.972
DENSENET	0.964	0.990	0.980

Table 2 summarizes the results, where we also report the (empirical) success probability achieved by an expert solving the (original) multiclass task in their own. We find that, by allowing for recommended subsets of varying size, our system is consistently superior to the top-k set-valued predictor. Moreover, we also find it very encouraging that, although the classifiers are highly accurate, our results suggest that an expert using our system can solve the prediction task significantly more accurately than the classifiers. More specifically, the relative reduction in misclassification probability goes from 72.2% (DenseNet) to 81.9% (ResNet-110). Finally, by using our system, our results suggest that the (average) expert would reduce their misclassification probability by 80%.

Robustness to violations of the IIA assumption in the estimator of the expert's success probability. To study the robustness of our system to violations of the IIA assumption in the estimator of the expert's success probability, we use the same experimental setting as in the synthetic experiments, where the parameter p controls the severity of the violation of the IIA assumption at test time. Figure 4

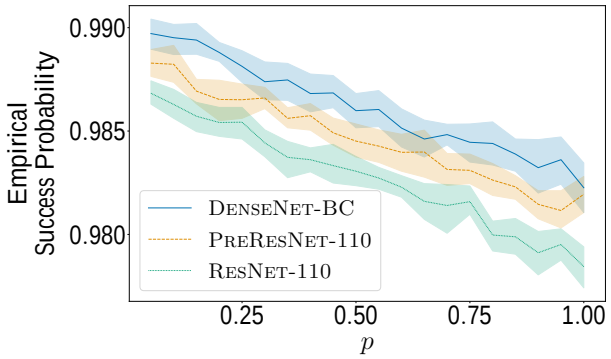


Figure 4. Empirical success probability achieved by an expert using our system with three different classifiers during test against severity p of the violation of the IIA assumption on the CIFAR-10H dataset. The empirical success probability achieved by the expert at solving the (original) multiclass task during test is $\mathsf{P}[\hat{Y} = Y \,;\, \mathcal{Y}]$ 0.947. The size of the calibration and estimation sets is m = 1,500. Shaded regions correspond to 95% confidence intervals.

summarizes the results for different p values. It is remarkable that, even for highly accurate classifiers like the ones used for our experiments, the expert benefits from using our system even when $p = 1$. This is because, for accurate classifiers, many prediction sets are singletons containing the true label, as shown in Appendix H.

7. Conclusions

We have initiated the development of automated decision support systems that, by design, do not require human experts to understand when each of their recommendations is accurate to improve their performance with high probability. We have focused on multiclass classification and designed a system that, for each data sample, recommends a subset of labels to the experts using a classifier. Moreover, we have shown that our system can help experts make predictions more accurately and is robust to the accuracy of the classifier and the estimator of the expert's success probability.

Our work opens up many interesting avenues for future work. For example, we have considered a simple, well-known conformal score function from the literature. However, it would be valuable to develop score functions especially designed for decision support systems. Moreover, it would be interesting to perform online estimation of the expert's conditional success probability. Further, it would be important to investigate the ethical impact of our system, including human trust and bias, understand the robustness of our system to malicious attacks, and consider alternative performance metrics such as expert prediction time. Finally, it would be important to deploy and evaluate our system on a real-world application with human experts.

Acknowledgements

We would like to thank the anonymous reviewers for constructive feedback, which has helped improve our paper. Gomez-Rodriguez acknowledges support from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 945719). Wang acknowledges support from NSF Awards IIS-1901168 and IIS-2008139. All content represents the opinion of the authors, which is not necessarily shared or endorsed by their respective employers and/or sponsors.

References

Angelopoulos, A., Bates, S., Malik, J., and Jordan, M. I. Uncertainty sets for image classifiers using conformal prediction. In *International Conference on Learning Representations*, 2021.

Angelopoulos, A. N. and Bates, S. A gentle introduction to conformal prediction and distribution-free uncertainty quantification. *arXiv preprint arXiv:2107.07511*, 2021.

Babbar, V., Bhatt, U., and Weller, A. On the utility of prediction sets in human-ai teams. *arXiv preprint arXiv:2205.01411*, 2022.

Balasubramanian, V., Ho, S.-S., and Vovk, V. *Conformal prediction for reliable machine learning: theory, adaptations and applications*. Newnes, 2014.

Bansal, G., Nushi, B., Kamar, E., Lasecki, W. S., Weld, D. S., and Horvitz, E. Beyond accuracy: The role of mental models in human-ai team performance. *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, 7(1):2–11, Oct. 2019. URL https://ojs.aaai.org/index.php/HCOMP/article/view/5285.

Bates, S., Angelopoulos, A., Lei, L., Malik, J., and Jordan, M. Distribution-free, risk-controlling prediction sets. *J. ACM*, 68(6), sep 2021. ISSN 0004-5411. doi: 10.1145/3478535. URL https://doi.org/10.1145/3478535.

Beach, L. R. Broadening the definition of decision making: The role of prechoice screening of options. *Psychological Science*, 4(4):215–220, 1993.

Ben-Akiva, M. and Boccara, B. Discrete choice models with latent choice sets. *International Journal of Research in Marketing*, 12(1):9–24, 1995.

Bordt, S. and von Luxburg, U. When humans and machines make joint decisions: A non-symmetric bandit model. *arXiv preprint arXiv:2007.04800*, 2020.

Chzhen, E., Denis, C., Hebiri, M., and Lorieul, T. Set-valued classification–overview via a unified framework. *arXiv preprint arXiv:2102.12318*, 2021.

De, A., Koley, P., Ganguly, N., and Gomez-Rodriguez, M. Regression under human assistance. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.

De, A., Okati, N., Zarezade, A., and Gomez-Rodriguez, M. Classification under human assistance. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021.

Del Coz, J. J., Díez, J., and Bahamonde, A. Learning nondeterministic classifiers. *Journal of Machine Learning Research*, 10(10), 2009.

Grgić-Hlača, N., Engel, C., and Gummadi, K. P. Human decision making with machine assistance: An experiment on bailing and jailing. *Proceedings of the ACM on Human-Computer Interaction*, 3(CSCW):1–25, 2019.

Gupta, C., Podkopaev, A., and Ramdas, A. Distribution-free binary classification: prediction sets, confidence intervals and calibration. In *Advances in Neural Information Processing Systems*, 2020.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016a.

He, K., Zhang, X., Ren, S., and Sun, J. Identity mappings in deep residual networks. In *European conference on computer vision*, pp. 630–645. Springer, 2016b.

Heiss, F. *Discrete choice methods with simulation*. Taylor & Francis, 2016.

Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. Densely connected convolutional networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2261–2269, 2017. doi: 10.1109/CVPR.2017.243.

Huber, J., Payne, J. W., and Puto, C. Adding asymmetrically dominated alternatives: Violations of regularity and the similarity hypothesis. *Journal of consumer research*, 9 (1):90–98, 1982.

Hulsman, R. Distribution-free finite-sample guarantees and split conformal prediction. *arXiv preprint arXiv:2210.14735*, 2022.

Jiao, W., Atwal, G., Polak, P., Karlic, R., Cuppen, E., Danyi, A., de Ridder, J., van Herpen, C., Lolkema, M. P., Steeghs, N., et al. A deep learning system accurately classifies primary and metastatic cancers using passenger mutation patterns. *Nature communications*, 11(1):1–12, 2020.

Kerrigan, G., Smyth, P., and Steyvers, M. Combining human predictions with model probabilities via confusion matrices and calibration. *arXiv preprint arXiv:2109.14591*, 2021.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. *Citeseer*, 2009.

Levy, A., Agrawal, M., Satyanarayan, A., and Sontag, D. Assessing the impact of automated suggestions on decision making: Domain experts mediate model errors but take less initiative. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, 2021.

Liu, R., Rizzo, S., Whipple, S., Pal, N., Pineda, A. L., Lu, M., Arnieri, B., Lu, Y., Capra, W., Copping, R., et al. Evaluating eligibility criteria of oncology trials using real-world data and ai. *Nature*, 592(7855):629–633, 2021.

Liu, Z.-G., Pan, Q., Dezert, J., and Mercier, G. Credal classification rule for uncertain data based on belief functions. *Pattern Recognition*, 47(7):2532–2541, 2014.

Lubars, B. and Tan, C. Ask not what ai can do, but what ai should do: Towards a framework of task delegability. *Advances in Neural Information Processing Systems*, 32: 57–67, 2019.

Luce, R. D. On the possible psychophysical laws. *Psychological review*, 66(2):81, 1959.

Ma, L. and Denoeux, T. Partial classification in the belief function framework. *Knowledge-Based Systems*, 214: 106742, 2021.

Meresht, V. B., De, A., Singla, A., and Gomez-Rodriguez, M. Learning to switch among agents in a team. *Transactions on Machine Learning Research*, 2022.

Mortier, T., Wydmuch, M., Dembczyński, K., Hüllermeier, E., and Waegeman, W. Efficient set-valued prediction in multi-class classification. *Data Mining and Knowledge Discovery*, 35(4):1435–1469, 2021.

Mozannar, H. and Sontag, D. Consistent estimators for learning to defer to an expert. In *International Conference on Machine Learning*, pp. 7076–7087, 2020.

Nguyen, V.-L. and Hüllermeier, E. Multilabel classification with partial abstention: Bayes-optimal prediction under label independence. *Journal of Artificial Intelligence Research*, 72:613–665, 2021.

Nourani, M., King, J. T., and Ragan, E. D. The role of domain expertise in user trust and the impact of first impressions with intelligent systems. *ArXiv*, abs/2008.09100, 2020.

Okati, N., De, A., and Gomez-Rodriguez, M. Differentiable learning under triage. In *Advances in Neural Information Processing Systems*, 2021.

Papenmeier, A., Englebienne, G., and Seifert, C. How model accuracy and explanation fidelity influence user trust. *arXiv preprint arXiv:1907.12652*, 2019.

Peterson, J. C., Battleday, R. M., Griffiths, T. L., and Russakovsky, O. Human uncertainty makes classification more robust. *arXiv preprint arXiv:1908.07086*, 2019.

Podkopaev, A. and Ramdas, A. Distribution-free uncertainty quantification for classification under label shift. In *Proceedings of the 37th Conference on Uncertainty in Artificial Intelligence*, 2021.

Raghu, M., Blumer, K., Corrado, G., Kleinberg, J., Obermeyer, Z., and Mullainathan, S. The algorithmic automation problem: Prediction, triage, and human effort. *arXiv preprint arXiv:1903.12220*, 2019.

Romano, Y., Patterson, E., and Candes, E. Conformalized quantile regression. *Advances in Neural Information Processing Systems*, 32:3543–3553, 2019.

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

Simonson, I. Choice based on reasons: The case of attraction and compromise effects. *Journal of consumer research*, 16(2):158–174, 1989.

Straitouri, E., Singla, A., Meresht, V. B., and Gomez-Rodriguez, M. Reinforcement learning under algorithmic triage. *arXiv preprint arXiv:2109.11328*, 2021.

Suresh, H., Lao, N., and Liccardi, I. Misplaced trust: Measuring the interference of machine learning in human decision-making. In *12th ACM Conference on Web Science*, pp. 315–324, 2020.

Tversky, A. Elimination by aspects: A theory of choice. *Psychological review*, 79(4):281, 1972.

Tversky, A. and Simonson, I. Context-dependent preferences. *Management Science*, 39(10):1179–1189, 1993. ISSN 00251909, 15265501. URL http://www.jstor.org/stable/2632953.

Vodrahalli, K., Gerstenberg, T., and Zou, J. Uncalibrated models can improve human-ai collaboration. In *Advances in Neural Information Processing Systems*, 2022.

Vovk, V., Gammerman, A., and Shafer, G. *Algorithmic Learning in a Random World*. Springer-Verlag, Berlin, Heidelberg, 2005. ISBN 0387001522.

Wang, L., Joachims, T., and Gomez-Rodriguez, M. Improving screening processes via calibrated subset selection. In *Proceedings of the 39th International Conference on Machine Learning*, 2022.

Wang, X. and Yin, M. Are explanations helpful? a comparative study of the effects of explanations in ai-assisted decision-making. In *26th International Conference on Intelligent User Interfaces*, pp. 318–328, 2021.

Wright, P. and Barbour, F. *Phased decision strategies: Sequels to an initial screening*. Graduate School of Business, Stanford University, 1977.

Yang, G., Destercke, S., and Masson, M.-H. Cautious classification with nested dichotomies and imprecise probabilities. *Soft Computing*, 21(24):7447–7462, 2017.

Yin, M., Wortman Vaughan, J., and Wallach, H. Understanding the effect of accuracy on trust in machine learning models. In *Proceedings of the 2019 chi conference on human factors in computing systems*, pp. 1–12, 2019.

Zhang, Y., Liao, Q. V., and Bellamy, R. K. Effect of confidence and explanation on accuracy and trust calibration in ai-assisted decision making. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pp. 295–305, 2020.

A. Proofs

A.1. Proof of Proposition 4.1

Given the estimators $\hat{\alpha}$ of $\mathsf{P}[\hat{Y} = Y ; C]$, we have that, for each $\alpha \in A$, it holds that

$$\left| \hat{\alpha} \quad \mathsf{P}[\hat{Y} = Y ; C] \right| \quad \epsilon_{;=m} \tag{11}$$

with probability at least $1 \quad \delta/m$. By applying the union bound, we know that the above events hold simultaneously for all $\alpha \in A$ with probability at least $1 \quad \delta$. Moreover, by rearranging, the above expression can be rewritten as

$$\hat{\alpha} \quad \epsilon_{;=m} \quad \mathsf{P}[\hat{Y} = Y ; C] \quad \hat{\alpha} + \epsilon_{;=m}. \tag{12}$$

Let $\hat{\alpha} = \operatorname{argmax}_{\alpha \in A} f(\hat{\alpha} \quad \epsilon_{;=m})g$. For $\hat{\alpha}$, with probability $1 \quad \delta$, it holds that for all $\alpha \in A$,

$$\begin{aligned}
\mathsf{P}[\hat{Y} = Y ; C_{\hat{\alpha}}] \quad \hat{\alpha}_{\hat{\alpha}} \quad \epsilon_{\hat{\alpha};=m} \quad \hat{\alpha} \quad \epsilon_{;=m} &= \hat{\alpha} \quad \epsilon_{;=m} + 2\epsilon_{;=m} \quad 2\epsilon_{;=m} \\
&= \hat{\alpha} + \epsilon_{;=m} \quad 2\epsilon_{;=m}, \\
&\quad \mathsf{P}\left[\hat{Y} = Y; C\right] \quad 2\epsilon_{;=m},
\end{aligned}$$

where the last inequality follows from Eq. 12.

A.2. Derivation of Error Expression for Hoeffding's Inequality

From Hoeffding's inequality we have that:

Theorem A.1. *Let $Z_1, ..., Z_k$ be i.i.d., with $Z_i \in [a, b], i = 1, ..., k, a < b$ and $\hat{\alpha}$ be the empirical estimate $\hat{\alpha} = \frac{\sum_{i=1}^{k} Z_i}{k}$ of $\mathsf{E}[Z] = \mathsf{E}[Z_i]$. Then:*

$$\mathsf{P}\left[\hat{\alpha} \quad \mathsf{E}[Z] \quad \epsilon\right] \quad \exp\left(\frac{2k\epsilon^2}{(b \quad a)^2} \right) \tag{13}$$

and

$$\mathsf{P}\left[\hat{\alpha} \quad \mathsf{E}[Z] \quad \epsilon\right] \quad \exp\left(\frac{2k\epsilon^2}{(b \quad a)^2} \right) \tag{14}$$

hold for all $\epsilon \quad 0$.

In our case we have $k = m$ and $Z_i = \mathbb{I}\{Y_i \in C_{(X_i)}\}\mathsf{P}\left[\hat{Y} = Y_i; C \mid Y_i \in C_{(X_i)}\right] \in (0, 1)$. Moreover, note that the expectation of Z_i is given by:

$$\begin{aligned}
\mathsf{E}[Z_i] &= \mathsf{E}\left[\mathbb{I}\{Y_i \in C_{(X_i)}\}\mathsf{P}\left[\hat{Y} = Y_i; C \mid Y_i \in C_{(X_i)}\right]\right] \\
&= \mathsf{E}\left[\mathbb{I}\{Y_i \in C_{(X_i)}\}\mathsf{P}\left[\hat{Y} = Y_i; C \mid Y_i \in C_{(X_i)}\right] + \mathbb{I}\{Y_i \notin C_{(X_i)}\}\mathsf{P}\left[\hat{Y} = Y_i; C \mid Y_i \notin C_{(X_i)}\right]\right] \\
&= \mathsf{E}\left[\mathsf{P}\left[\hat{Y} = Y_i; C\right]\right] \\
&= \mathsf{P}\left[\hat{Y} = Y_i; C\right],
\end{aligned}$$

where the expectations are over the joint distribution of prediction sets $C_{(X)}$ and true labels Y.

Hence, for the empirical estimate $\hat{\alpha} = \hat{\alpha}$ of $\mathsf{P}[\hat{Y} = Y ; C]$ and its error $\epsilon = \epsilon_{;}$:

$$\mathsf{P}\left[\hat{\alpha} \quad \mathsf{P}[\hat{Y} = Y ; C] \quad \epsilon_{;}\right] \quad \exp\left(\frac{2m\epsilon^2_{;}}{(1 \quad 0)^2} \right) \tag{15}$$

and

$$\mathsf{P}\left[\hat{p} \quad \mathsf{P}[\hat{Y} = Y \,;\, C\,] \quad \epsilon_{\,;}\right] \quad \exp\left(\frac{2m\epsilon^2_{\,;}}{(1 \quad 0)^2}\right) \tag{16}$$

hold. Further, if we set

$$\delta = \exp\left(\quad 2m\epsilon^2_{\,;}\right), \tag{17}$$

then

$$1 \quad \mathsf{P}\left[\hat{p} \quad \mathsf{P}[\hat{Y} = Y \,;\, C\,] \quad \epsilon_{\,;}\right] \quad \delta \;) \; \mathsf{P}\left[\hat{p} \quad \mathsf{P}[\hat{Y} = Y \,;\, C\,] \quad \epsilon_{\,;}\right] \quad 1 \quad \delta \tag{18}$$

and

$$1 \quad \mathsf{P}\left[\hat{p} \quad \mathsf{P}[\hat{Y} = Y \,;\, C\,] \quad \epsilon_{\,;}\right] \quad \delta \;) \; \mathsf{P}\left[\hat{p} \quad \mathsf{P}[\hat{Y} = Y \,;\, C\,] \quad \epsilon_{\,;}\right] \quad 1 \quad \delta \tag{19}$$

hold for any $\epsilon_{\,;} \quad 0$. As follows, based on Eq. 17:

$$\delta = \exp\left(\quad 2m\epsilon^2_{\,;}\right) \;) \; \log\frac{1}{\delta} = 2m\epsilon^2_{\,;} \;) \; \epsilon^2_{\,;} = \frac{\log\frac{1}{\delta}}{2m} \;) \; \epsilon_{\,;} = \sqrt{\frac{\log\frac{1}{\delta}}{2m}}.$$

A.3. Proof of Proposition D.1

We proceed similarly as in the Appendix A.5 in Hulsman (2022). First, note that, by definition, we have that

$$\hat{q}_{\,1} = s_{(d(1 \quad \alpha_1)(m+1)e)} \quad \text{and} \quad \hat{q}_{\,2} = s_{(d(1 \quad \alpha_2)(m+1)e)},$$

where $s_{(i)}$ denotes the i-th smallest conformal score in the calibration set D_{cal}. Then, as long as the conformal scores in the calibration set are almost surely distinct, it follows directly from Proposition 4 in Hulsman (2022) that

$$\mathsf{P}\left[\hat{q}_{\,2} < s(X,Y) \quad \hat{q}_{\,1} \,|\, D_{\text{cal}}\right] \quad \text{Beta}(l, m \quad l + 1), \tag{20}$$

where $l = d(m + 1)(1 \quad \alpha_1)e \quad d(m + 1)(1 \quad \alpha_2)e$. Moreover, for any $(X,Y) \quad P(X)P(Y \,|\, X)$, we have that, by construction, $Y \in C_{\,1;\,2}(X)$ if and only if $s(X,Y) \in (\hat{q}_{\,2}, \hat{q}_{\,1})$. Then, Eq. 22 follows directly from Eq. 20.

B. Implementation Details

To implement our algorithms and run all the experiments on synthetic and real data, we used PyTorch 1.12.1, NumPy 1.20.1 and Scikit-learn 1.0.2 on Python 3.9.2. For reproducibility, we use a fixed random seed in all random procedures. Moreover, we set $\delta = 0.1$ everywhere.

Synthetic prediction tasks. We create $4 \quad 3 = 12$ different prediction tasks, where we vary the number of labels $n \; 2 \; f10, 50, 100g$ and the level of difficulty of the task. More specifically, for each value of n, we create four different tasks of increasing difficulty where the success probability of the logistic regression classifier is $P[Y^0 = Y] = 0.9, 0.7, 0.5$ and 0.3, respectively.

To create each task, we use the function `make_classification` of the Scikit-learn library. This function allows the creation of data for synthetic prediction tasks with very particular user-defined characteristics, through the generation of clusters of normally distributed points on the vertices of a multidimensional hypercube. The number of the dimensions of the hypercube indicates the number of informative features of each sample, which in our case we set at 15 for all prediction tasks. Linear combinations of points, *i.e.*, the informative features, are used to create redundant features, the number of which we set at 5. The difficulty of the prediction task is controlled through the size of the hypercube, with a multiplicative factor, namely `clas_sep`, which we tuned accordingly for each value n so that the success probability of the logistic regression classifier above spans a wide range of values across tasks. All the selected values of this parameter can be found in the configuration file `config.py` in the code. Finally, we set the proportion of the samples assigned to each label, *i.e.*, the function parameter `weights`, using a Dirichlet distribution of order n with parameters $\alpha_1 = ... = \alpha_n = 1$.

C. Additional Synthetic Prediction Tasks, Number of Labels and Amount of Calibration and Estimation Data

To complement the results in Table 1 in the main paper, we experiment with additional prediction tasks with different number of labels n and amount of calibration and estimation data m. For each value of n and m, we estimate the success probability $\mathsf{P}[\hat{Y} = Y \,;\, C_\wedge]$ achieved by four different experts using our system, each with a different success probability $\mathsf{P}[\hat{Y} = Y \,;\, Y]$, on four prediction tasks where the classifier achieves a different success probability $\mathsf{P}[Y^0 = Y]$. Figure 5 summarizes the results.

$\mathsf{P}[\hat{Y} = Y \,;\, Y]$	$\mathsf{P}[Y^0 = Y]$ 0.3	0.5	0.7	0.9
0.3	0.56	0.72	0.84	0.94
0.5	0.68	0.80	0.89	0.95
0.7	0.79	0.87	0.93	0.97
0.9	0.92	0.95	0.97	0.99

(a) $n = 50$, $m = 1{,}200$

$\mathsf{P}[\hat{Y} = Y \,;\, Y]$	$\mathsf{P}[Y^0 = Y]$ 0.3	0.5	0.7	0.9
0.3	0.62	0.76	0.87	0.95
0.5	0.72	0.83	0.91	0.96
0.7	0.83	0.90	0.95	0.98
0.9	0.93	0.96	0.98	0.99

(b) $n = 100$, $m = 1{,}200$

$\mathsf{P}[\hat{Y} = Y \,;\, Y]$	$\mathsf{P}[Y^0 = Y]$ 0.3	0.5	0.7	0.9
0.3	0.42	0.58	0.75	0.91
0.5	0.55	0.66	0.80	0.93
0.7	0.72	0.79	0.87	0.96
0.9	0.90	0.92	0.94	0.98

(c) $n = 10$, $m = 400$

$\mathsf{P}[\hat{Y} = Y \,;\, Y]$	$\mathsf{P}[Y^0 = Y]$ 0.3	0.5	0.7	0.9
0.3	0.56	0.73	0.84	0.94
0.5	0.67	0.80	0.88	0.96
0.7	0.79	0.88	0.93	0.98
0.9	0.92	0.94	0.97	0.99

(d) $n = 50$, $m = 400$

$\mathsf{P}[\hat{Y} = Y \,;\, Y]$	$\mathsf{P}[Y^0 = Y]$ 0.3	0.5	0.7	0.9
0.3	0.62	0.77	0.87	0.95
0.5	0.73	0.83	0.91	0.97
0.7	0.83	0.89	0.95	0.98
0.9	0.93	0.96	0.98	0.99

(e) $n = 100$, $m = 400$

Figure 5. Empirical success probability achieved by four different experts using our system during test, each with a different success probability $\mathsf{P}[\hat{Y} = Y \,;\, Y]$, on four prediction tasks where the classifier achieves a different success probability $\mathsf{P}[Y^0 = Y]$. Each table corresponds to a different number of label values n and calibration and estimation set size m. For readability, each cell shows only the average since the standard errors are all below 10^{-2}.

D. Beyond Standard Conformal Prediction

In Section 4, we have used standard conformal prediction (Angelopoulos & Bates, 2021) to construct the recommended subsets $C(X)$—we have constructed $C(X)$ by comparing the conformal scores $s(X, y)$ to a single threshold $\hat{q}$, as shown in Eq. 3. Here, we introduce a set-valued predictor based on conformal prediction that constructs $C(X)$ using two thresholds $\hat{q}_{\alpha_1}$ and $\hat{q}_{\alpha_2}$. By doing so, the recommended subsets will include label values whose corresponding conformal scores are neither unreasonably large, as in standard conformal prediction, nor unreasonably low in comparison with the conformal scores of the samples in the calibration set D_{cal}. This may be useful in scenarios where the classifier underpinning our system has not particularly high average accuracy[16].

More specifically, given a calibration set $D_{\text{cal}} = \{(x_i, s_i)\}_{i=1}^{m}$, let $\alpha_1, \alpha_2 \in [0, 1]$, with $\alpha_1 < \alpha_2$, and $\hat{q}_{\alpha_1}$ and $\hat{q}_{\alpha_2}$ be the $\frac{\lceil (m+1)(1-\alpha_1) \rceil}{m}$ and $\frac{\lceil (m+1)(1-\alpha_2) \rceil}{m}$ empirical quantiles of the conformal scores $s(x_1, y_1), \ldots, s(x_m, y_m)$. If we construct the subsets $C_{\alpha_1, \alpha_2}(X)$ for new data samples as follows:

$$C_{\alpha_1, \alpha_2}(X) = \{y \mid \hat{q}_{\alpha_2} < s(X, y) \leq \hat{q}_{\alpha_1}\}, \tag{21}$$

we have that the probability that the true label Y belongs to the subset $C_{\alpha_1, \alpha_2}(X)$ conditionally on the calibration set D_{cal} is almost exactly $\alpha_2 - \alpha_1$ with high probability as long as the size m of the calibration set is sufficiently large. More specifically, we first note that the coverage probability is a random quantity whose distribution is given by the following proposition, which is the counterpart of Proposition 3.1:

Proposition D.1. *For a decision support system C_{α_1, α_2} that constructs $C_{\alpha_1, \alpha_2}(X)$ using Eq. 21, as long as the conformal scores $s(x_i, y_i)$ for all $(x_i, y_i) \in D_{cal}$ are almost surely distinct, it holds that:*

$$\mathrm{P}[Y \in C_{\alpha_1, \alpha_2}(X) \mid D_{cal}] \sim \mathit{Beta}(l, m - l + 1), \tag{22}$$

where $l = \lceil (m+1)(1-\alpha_1) \rceil - \lceil (m+1)(1-\alpha_2) \rceil$.

As an immediate consequence of Proposition D.1, using the definition of the beta distribution, we have that

$$\alpha_2 - \alpha_1 - \mathrm{E}[\mathrm{P}[Y \in C_{\alpha_1, \alpha_2}(X) \mid D_{\text{cal}}]] = \alpha_2 - \alpha_1 + \frac{c_1 - c_2}{m+1} - \alpha_2 - \alpha_1 + \frac{1}{m+1},$$

where $c_1, c_2 \in [0, 1)$. Moreover, given a target probability $\alpha_2 - \alpha_1$ and tolerance values $\delta, \epsilon \in (0, 1)$, we can compute the minimum size m of the calibration set D_{cal} such that C_{α_1, α_2} enjoys Probably Approximately Correct (PAC) coverage guarantees, *i.e.*, with probability $1 - \delta$, it holds that

$$\alpha_2 - \alpha_1 - \epsilon - \mathrm{P}[Y \in C_{\alpha_1, \alpha_2}(X) \mid D_{\text{cal}}] - \alpha_2 - \alpha_1 + \epsilon.$$

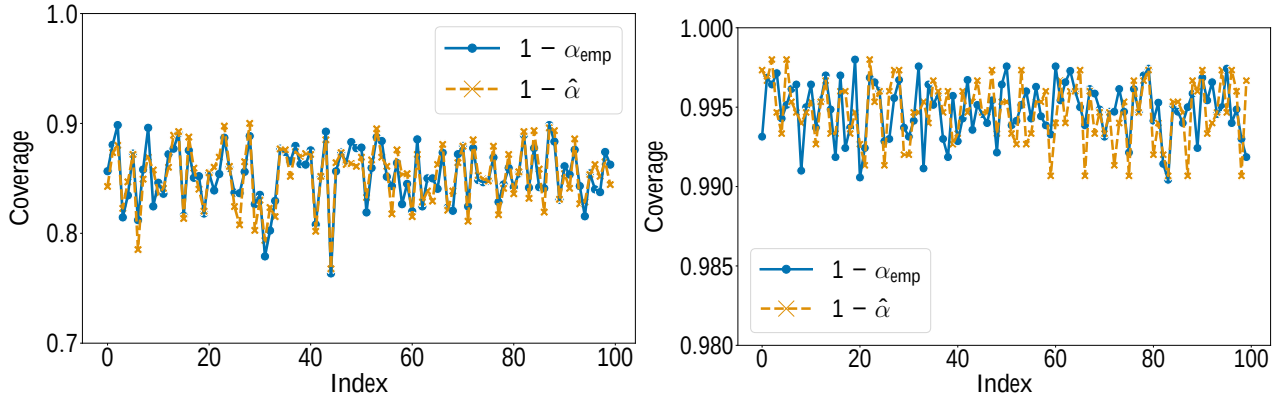
Finally, given an estimator of the expert's success probability $\hat{\mu}_{\alpha_1, \alpha_2}$ such that for each $\alpha_1 < \alpha_2$ and $\delta \in (0, 1)$, with probability at least $1 - \delta$, it holds that $|\hat{\mu}_{\alpha_1, \alpha_2} - \mathrm{P}[\hat{Y} = Y \mid C_{\alpha_1, \alpha_2}]| \leq \epsilon_{\alpha_1, \alpha_2, \delta}$, we can proceed similarly as in standard conformal prediction to find the near optimal $\hat{\alpha}_1, \hat{\alpha}_2 \in A$ that maximizes the expert's success probability with high probability, by using $\hat{\mu}_{\alpha_1, \alpha_2}$ and $\epsilon_{\alpha_1, \alpha_2, \delta/2} = \delta/(m(m-1))$. Here, it is worth pointing out that, in contrast with the case of standard conformal prediction, the time complexity of finding the near optimal $\hat{\alpha}_1$ and $\hat{\alpha}_2$ is $O(m \log m + mn \log n + mn^2)$. Moreover, we can still rely on the practical method to estimate the expert's conditional success probability introduced in Section 4.

[16] In such scenarios, the conformal scores of the samples in the calibration set can occasionally have low values—otherwise, the classifier would be highly accurate—and thus it is beneficial to exclude label values with (very) low conformal scores from the recommended subsets—those label values the classifier is confidently wrong about.

E. Sensitivity to the Choice of Calibration Set

In this section, we repeat the experiments on synthetic and real data using 100 independent realizations of the calibration, estimation and test sets. Then, for each data split, we compare the empirical coverage $\frac{1}{|D_{\text{test}}|}\sum_{(x,y)\in D_{\text{set}}} \mathbb{I}[y \in C_{\hat{}}(x)]$:= $1 - \hat{\alpha}_{\text{emp}}$ achieved by our system $C_{\hat{}}$ on the test set D_{test} to the corresponding target coverage $1 - \hat{\alpha}$.

Figure 6 summarizes the results for (a) one synthetic prediction task and one synthetic expert and (b) one popular deep neural network classifier on the CIFAR-10H dataset. We find that the value of the near-optimal $\hat{\alpha}$ does not vary significantly across experiments (*i.e.*, across calibration sets) and, for each experiment, the empirical coverage $1 - \hat{\alpha}_{\text{emp}}$ is very close to and typically higher than the target coverage $1 - \hat{\alpha}$. We found similar results for other expert-classifier pairs with different success probabilities.



(a) Synthetic Task

(b) CIFAR-10H

Figure 6. Empirical test coverage $1 - \alpha_{\text{emp}}$ and target coverage $1 - \hat{\alpha}$ for 100 independent realizations of the calibration, estimation and test sets. In Panel (a), the synthetic task comprises a classifier with $\mathbb{P}[Y^0 = Y] = 0.5$ and an expert with $\mathbb{P}[\hat{Y} = Y \mid Y] = 0.5$, the number of labels is $n = 10$ and the size of the calibration and estimation sets is $m = 1{,}200$. In Panel (b), the classifier is the popular DenseNet classifier and $m = 1{,}500$.

F. Success Probability Achieved by an Expert using Top-k Set-Valued Predictors

In this section, we estimate the success probability achieved by an expert using the top-k set-valued predictor for different k values using both synthetic and real data. Figures 7 and 8 summarize the results, which show that, by allowing for recommended subsets of varying size, our system is consistently superior to the top-k set-valued predictor across configurations. Moreover, the results on synthetic data also show that, the higher the expert's success probability $\mathsf{P}[\hat{Y} = Y \,;\, Y]$, the greater the optimal k value (*i.e.*, the greater the optimal size of the recommended subsets $C_k(X)$). This latter observation is consistent with the behavior exhibited by our system, where the higher the expert's success probability $\mathsf{P}[\hat{Y} = Y \,;\, Y]$, the lower the value of the near-optimal $\hat{\alpha}$ and thus the greater the average size of the recommended subsets $C_{\hat{\alpha}}(X)$, as shown in Figure 9.

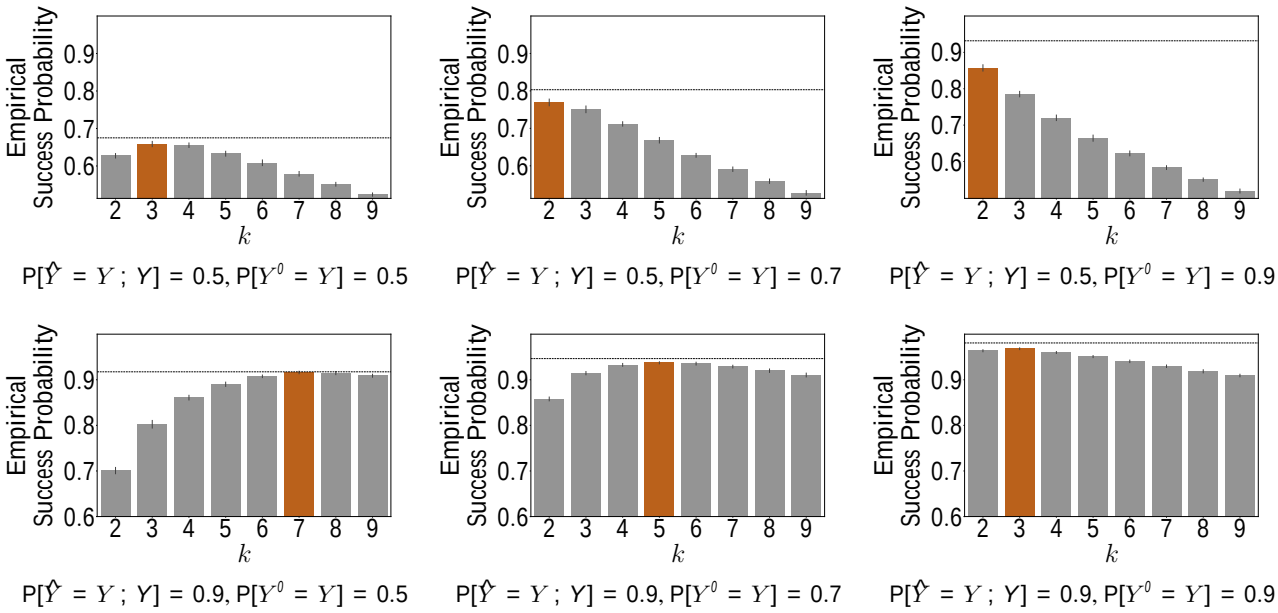


Figure 7. Empirical success probability achieved by two different experts using the top-k set-valued predictor (C_k) during test, each with a different success probability $\mathsf{P}[\hat{Y} = Y \,;\, Y]$, on three prediction tasks where the classifier achieves a different success probability $\mathsf{P}[Y^0 = Y]$. In each panel, the horizontal dashed line shows the empirical success probability achieved by the same experts using our system ($C_{\hat{\alpha}}$) during test. In all panels, the number of labels is $n = 10$, the size of the calibration and estimation sets is $m = 1{,}200$ and the results for the optimal k value during test are highlighted in orange.

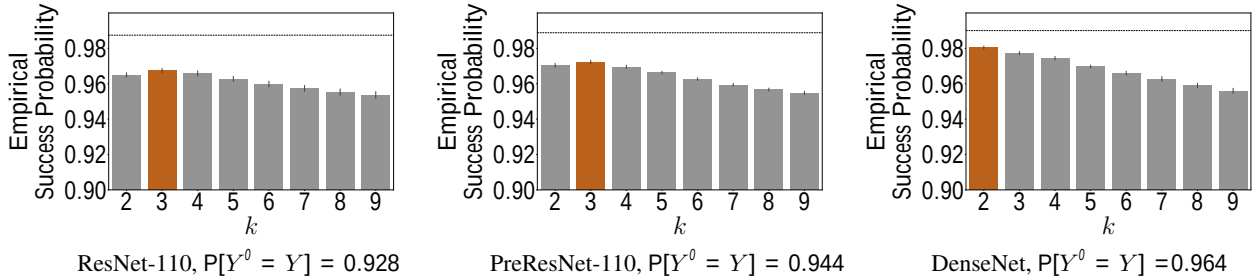


Figure 8. Empirical success probability achieved by an expert using three different top-k predictors (C_k) during test, each with a different deep neural network classifier, on the CIFAR-10H dataset. In each panel, the horizontal dashed line shows an empirical success probability achieved by the same expert using our system ($C_{\hat{\alpha}}$) during test. In all panels, the size of the calibration and estimation sets is $m = 1{,}500$ and the results for the optimal k value during test are highlighted in orange.

G. Size Distribution of the Recommended Subsets

Figure 9 shows the empirical size distribution of the subsets $C_\wedge(X)$ recommended by our system during test for different experts and prediction tasks on synthetic data. The results show that, as the expert's success probability $\mathrm{P}[\hat{Y} = Y\,;\, Y]$ increases and the near optimal $\hat{\alpha}$ decreases, the spread of the size distribution increases.

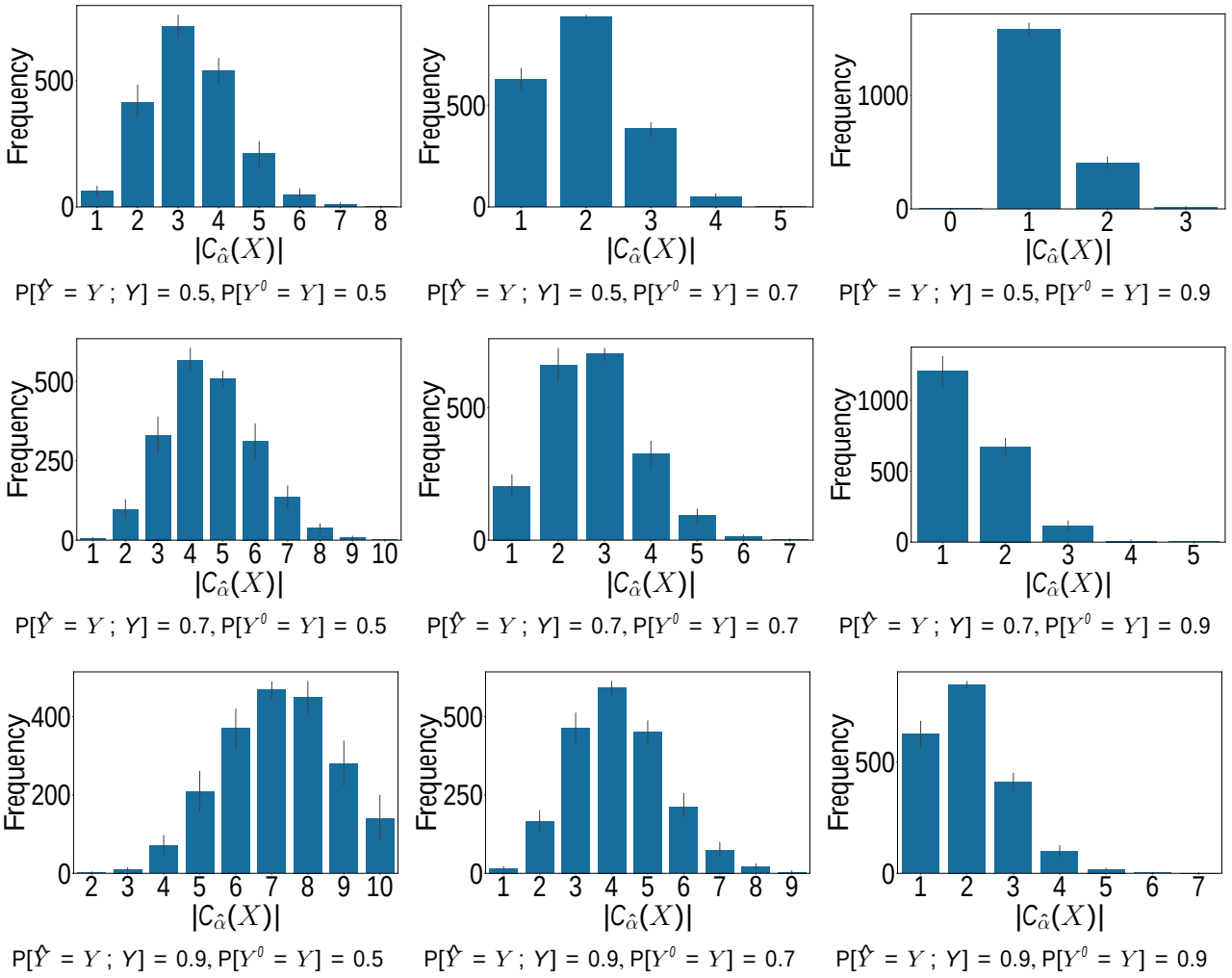
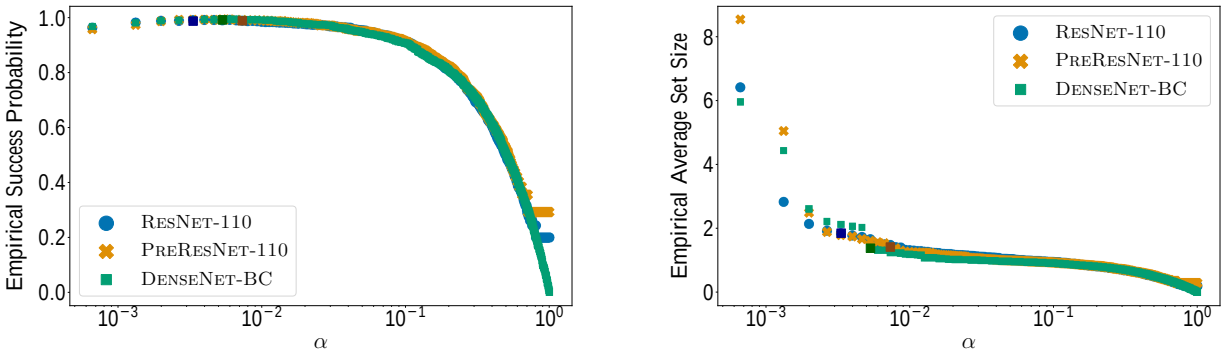


Figure 9. Empirical size distribution of the subsets $C_{\hat{\alpha}}(X)$ recommended by our system during test for different prediction tasks where the expert and the classifier achieve different success probabilities $\mathrm{P}[\hat{Y} = Y\,;\, Y]$ and $\mathrm{P}[Y^0 = Y]$, respectively. In all panels, the number of labels is $n = 10$ and the size of the calibration and estimation sets is $m = 1{,}200$.

H. Performance of Our System Under Different α Values

In this section, we complement the results on CIFAR-10H dataset in the main paper by comparing, for each choice of classifier, the performance of our system under the near optimal $\hat{\alpha}$ found by Algorithm 1 and under all other possible α values, including the optimal α . Figure 10 summarizes the results, which suggest that, similarly as in the experiments on synthetic data, the performance of our system under $\hat{\alpha}$ and α is very similar. However, since the classifiers are all highly accurate, the average size of the recommended subsets under $\hat{\alpha}$ and α is quite close to one even though $\hat{\alpha}$ is much smaller than in the experiments in synthetic data.



(a) Empirical success probability vs α

(b) Empirical average set size vs α

Figure 10. Empirical success probability achieved by an expert using our system and average size of the recommended sets during test for each α 2 A and for three popular deep neural network classifiers on the CIFAR-10H dataset. The size of the calibration and estimation sets is m = 1,500. Each marker corresponds to a different α value and the darker points correspond to $\hat{\alpha}$ for each task.

I. Additional Experiments using an Estimator of the Expert's Success Probability with a More Expressive Context

In this section, we repeat the experiments on the CIFAR-10H dataset using an alternative discrete choice model with a more expressive context which, additionally to the true label, distinguishes between different levels of difficulty across data samples. The goal here is to show that our results are not an artifact of the choice of context used in the main paper.

We consider three increasing levels of difficulty, denoted as L_{easy}, L_{medium}, L_{hard}. The difficulty levels correspond to the 50% and 25% quantiles of the experts' fractions of correct predictions per sample in the (original) multiclass classification task. Samples with a fraction of correct predictions larger than the 50% quantile belong to L_{easy}, those with a fraction of correct predictions smaller than the 25% quantile belong to L_{hard}, and the remaining ones belong to L_{medium}. Then, given a sample (x, y) of difficulty L, we assume that the expert's conditional success probability for the subset $C(x)$ is given by:

$$\mathsf{P}[\hat{Y} = y; C \mid y \in C(x), L] = \frac{e^{u^L_{yy}}}{\sum_{y' \in C(x)} e^{u^L_{yy'}}}, \tag{23}$$

where $u^L_{yy'}$ denotes the expert preference for the label value $y' \in Y$ whenever the true label is y and the difficulty level of the sample is L.

Further, to estimate the parameters $u^L_{yy'}$, we resort to the conditional confusion matrix for the expert predictions on samples of difficulty L, *i.e.*, $\mathbf{C}^L = \left[C^L_{yy'}\right]_{y,y' \in Y}$, where $C^L_{yy'} = \mathsf{P}[\hat{Y} = y'; Y \mid Y = y, L]$, and set $u^L_{yy'} = \log C^L_{yy'}$. Finally, we compute a Monte-Carlo estimate $\hat{\alpha}$ of the expert's success probability $\mathsf{P}[\hat{Y} = Y; C]$ required by Algorithm 1 using the above conditional success probability and an estimation set $D_{\text{est}} = \{(x_i, y_i)\}_{i \in [m]}$, *i.e.*,

$$\hat{\alpha} = \frac{1}{m} \sum_{i \in [m] \mid y_i \in C(x_i)} \mathsf{P}[\hat{Y} = y_i; C \mid y_i \in C(x_i), L(x_i)], \tag{24}$$

where $L(x_i) \in \{L_{\text{easy}}, L_{\text{medium}}, L_{\text{hard}}\}$ denotes the difficulty level of x_i.

Table 3 summarizes the results, which suggest that, in agreement with the main paper, an expert using our system may solve the prediction task significantly more accurately than the expert or the classifier on their own.

Table 3. Empirical success probabilities achieved by three popular deep neural network classifiers and by an expert using our system with these classifiers during test on the CIFAR-10H dataset. Here, we assume the expert follows the alternative discrete choice model defined by Eq. 23. The size of the calibration and estimation sets is $m = 1{,}500$ and the expert's empirical success probability at solving the (original) multiclass task is $\mathsf{P}[\hat{Y} = Y; Y] \approx 0.947$. Each cell shows only the average since the standard errors are all below 10^{-2}.

	CLASSIFIER	EXPERT USING $C_{\hat{\alpha}}$
RESNET-110	0.928	0.981
PRERESNET-110	0.944	0.983
DENSENET	0.964	0.987

Finally, similarly as in the main paper, we also found that our system performs well with a small amount of calibration and estimation data—the relative gain in empirical success probability achieved by an expert using our system with respect to the same expert on their own raises from $3.02 \pm 0.05\%$ under $m = 200$ to just $3.28 \pm 0.04\%$ under $m = 1{,}500$.