Machine Learning (2025) 114:202
https://doi.org/10.1007/s10994-025-06844-8

A calibration test for evaluating set-based epistemic uncertainty representations

Mira Jürgens[1] · Thomas Mortier[2] · Eyke Hüllermeier[3] · Viktor Bengs[3] · Willem Waegeman[1]

Received: 9 June 2025 / Revised: 14 July 2025 / Accepted: 15 July 2025 /
Published online: 5 August 2025

Abstract

The accurate representation of epistemic uncertainty is a challenging yet essential task in machine learning. A widely used representation corresponds to convex sets of probabilistic predictors, also known as credal sets. One popular way of constructing these credal sets is via ensembling or specialized supervised learning methods, where the epistemic uncertainty can be quantified through measures such as the set size or the disagreement among members. In principle, these sets should contain the true data-generating distribution. As a necessary condition for this validity, we adopt the strongest notion of calibration as a proxy. Concretely, we propose a novel statistical test to determine whether there is a convex combination of the set's predictions that is calibrated in distribution. In contrast to previous methods, our framework allows the convex combination to be instance-dependent, recognizing that different ensemble members may be better calibrated in different regions of the input space. Moreover, we learn this combination via proper scoring rules, which inherently optimize for calibration. Building on differentiable, kernel-based estimators of calibration errors, we introduce a nonparametric testing procedure and demonstrate the benefits of capturing instance-level variability on synthetic and real-world experiments.

Keywords Uncertainty estimation · Calibration · Ensembles · Credal sets · Epistemic uncertainty

1 Introduction

In supervised machine learning, it has become more and more important not only to have accurate predictors, but also to provide a reliable quantification of predictive uncertainty, i.e., the learner's uncertainty in the outcome $y \in \mathcal{Y}$ given a query instance $x \in \mathcal{X}$ for which a prediction is sought. Predictive uncertainty is often divided into *aleatoric* and *epistemic* uncertainty (Senge et al., 2014; Hüllermeier & Waegeman, 2021; Kendall & Gal, 2017; Gruber et al., 2023), where the former corresponds to uncertainty that cannot be reduced with further information (e.g. more training data), as it originates from inherent

Editors: Concha Bielza, Ana Carolina Lorena, Tiago Almeida.

Extended author information available on the last page of the article randomness in the relationship between features X and labels Y. Therefore, we assume that the *ground truth* is a conditional probability distribution $\mathbb{P}_{Y\,|\,X}$ on $\mathcal{Y}$, i.e. given an input sample $x \in \mathcal{X}$, each outcome y has a certain probability to occur, given by $\mathbb{P}_{Y\,|\,X=x}$. Even with perfect knowledge about the underlying data-generating process, the outcome cannot be predicted with certainty. However, in a typical machine learning scenario, the learner does not know $\mathbb{P}_{Y|X}$. Having a space of possible hypotheses, an estimator of the underlying probability distribution f within this space typically consists of a mapping $f : \mathcal{X} \rightarrow \mathbb{P}(\mathcal{Y})$, where $\mathcal{X}$ denotes the feature space and $\mathbb{P}(\mathcal{Y})$ the space of all probability distributions over the target space $\mathcal{Y}$. In essence, epistemic uncertainty refers to the uncertainty about the true $\mathbb{P}_{Y\,|\,X}$, or the "gap" between $\mathbb{P}_{Y\,|\,X}$ and f. One approach to represent this gap is via *second-order probability distributions*, assigning a probability for each of the first-order predicted probability distributions, i.e. the candidates for f. This is commonly done in Bayesian methods, such as Gaussian processes and Bayesian neural networks (Gelman et al., 2004), but also in evidential deep learning methods (Ulmer et al., 2023). However, the former usually involves computationally costly methods to approximate the intractable posterior distribution, while the latter has been criticised for producing unfaithful or unreliable representations of epistemic uncertainty (Bengs et al., 2022, 2023; Meinert et al., 2023; Juergens et al., 2024). An alternative way of representing epistemic uncertainty—which will be the topic of this paper—is through *sets* of probability distributions. Such sets are often referred to as *credal sets* (Walley, 1991) in the imprecise probability literature, and are commonly assumed to be closed and convex (Cozman, 2000). In essence, they are designed to represent ignorance, i.e. a lack of knowledge about the underlying ground truth, by not committing to one, but a set of potential probability distributions. They can be obtained in a direct manner, as via credal classifiers (Hüllermeier et al., 2022; Javanmardi et al., 2024; Wang et al., 2025; Caprio et al., 2024; Nguyen et al., 2022), which have recently gained popularity, or in an indirect manner, via various types of ensemble methods, such as bootstrapped ensembles, deep ensembles (Lakshminarayanan et al., 2017) and randomization techniques based on Monte Carlo dropout (Gal & Ghahramani, 2016). In credal sets, epistemic uncertainty is quantified via the size of the credal set (Sale et al., 2023) or the diversity of the corresponding ensemble (Lakshminarayanan et al., 2017).

Due to a lack of an objective ground-truth, epistemic uncertainty representations are often evaluated in an indirect manner, using downstream tasks such as out-of-distribution detection (Ovadia et al., 2019), robustness to adversarial attacks (Kopetzki et al., 2021), and active learning (Nguyen et al., 2022). However, recent studies have raised concerns about the usefulness of such tasks w.r.t. epistemic uncertainty evaluation (Abe et al., 2022; Meinert et al., 2023; Bengs et al., 2022). A crucial open question is whether existing representations of epistemic uncertainty can be interpreted in a statistically profound way. For the case of credal sets, one might wonder whether such representations are statistically *valid*, i.e., whether a credal set contains with high probability the true underlying conditional target distribution. One way to evaluate this validity criterion is via statistical tests, as exemplary visualized in Fig. 1. Chau et al. (2024) address this issue by proposing a two-sample test framework for credal sets. However, they assume having a sample of sufficient size from the same underlying conditional distribution. In the "classical" machine learning scenario that we look at, we do not have access to the ground truth conditional distribution nor to more than one realization (x_i, y_i) from it, making this direct evaluation infeasible.

One necessary condition for validity is *calibration*Niculescu-Mizil and Caruana (2005), which measures the consistency between predicted probabilities and actual frequencies. In this paper we will utilize in this paper the notion of distribution calibration (Vaicenavicius

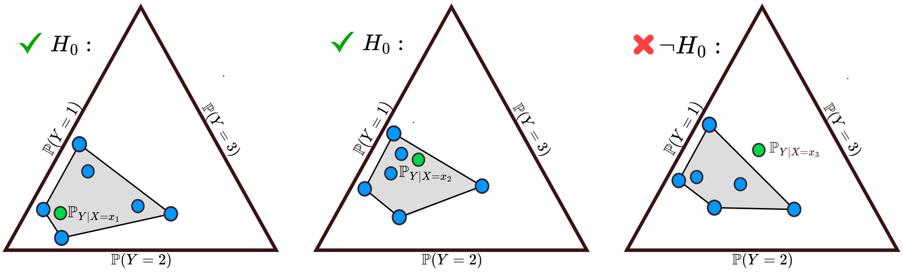


Fig. 1 Exemplary illustration of a desired statistical test of validity for the the case of $K = 3$ classes. Ideally, it should be *instance-dependent*, that is, allow the ground truth to be a convex combination with weights that vary across the instance space. In the left and middle panel the credal sets (composed by the blue dots) is *valid* because the true conditional distribution (green dot) lies inside the convex hull. In the right panel the true conditional distribution lies outside the convex hull, hence the null hypothesis should be *rejected*. In this work, we use calibration as an approximation to test validity, that is, we replace $\mathbb{P}_{Y|X}$ with $\mathbb{P}_{Y|f(X)}$ for a probabilistic predictor f

et al., 2019) as a surrogate method for validity. Failing calibration necessarily means failing validity, hence in this case one can be confident that the true data generating distribution is not contained inside the credal set. Hence, by introducing a calibration test for set-based epistemic uncertainty representations, we aim for a more direct evaluation than measuring the performance on downstream tasks.

Our work builds further upon the work of Mortier et al. (2023), who first proposed the usage of calibration as a proxy for epistemic uncertainty evaluation of credal sets. The test introduced in that study analysed whether, for a given convex set of probabilistic models, there exists a calibrated convex combination in the set. However, this existing test has important limitations, which motivate our current work. Specifically, the original test focuses on determining whether a *single* convex combination of ensemble predictions can achieve calibration, but it does so in an instance-agnostic manner. This limitation means that the approach does not account for the variability of model calibration in dependence on the instance space. Moreover, the previous test tries to simulate the distribution of the calibration error estimators under the null hypothesis via sampling predictions in the credal set at random, which possibly leads to an overestimation of how calibrated the found convex combination is. In this paper, we address these shortcomings by introducing an instance-dependent approach to calibration testing, with two important modifications: First, due to the instance-dependency of the underlying convex combination of probabilistic predictions, we consider situations where predictions of the set are “differently well” calibrated for different regions in the instance space. Second, by using an optimization-based instead of a sampling-based algorithm, we achieve a more reliable estimate of the calibration statistic under the null hypothesis. Together the two modifications result in better control of the statistical Type I error, while simultaneously increasing the power of the test.

The paper is organized as follows. In Sect. 2 we review the literature on calibration, and we formally discuss the calibration measures that are used further on. In Sect. 3 we introduce both the concept of instance-dependent validity as well as the (weaker) notion of calibration for credal sets. We then propose our novel, nonparametric calibration test that is able to assess whether an epistemic uncertainty representation via credal sets is calibrated. Furthermore, we introduce an optimization algorithm for finding the most calibrated convex combination. It includes training a neural network as a meta-learner, using proper scoring rules as loss functions. In Sect. 4 we empirically evaluate the statistical Type I and Type II error of our test in different scenarios, thereby showing its improvement over the test proposed by Mortier et al. (2023) We further demonstrate its usefulness on validating common ensemble-based epistemic uncertainty representations of models trained on real-world datasets.

2 Calibration of first-order predictors

We start with reviewing common calibration metrics that have been introduced for evaluating distribution calibration in multi-class classification. We also discuss differentiable estimators of these metrics, and statistical tests that assess calibration for a *single* probabilistic model. The extension to *sets* of probabilistic models will be discussed in Sect. 3.

2.1 General setting

We assume a multi-class classification setting with feature space $\mathcal{X} \subseteq \mathbb{R}^d$ and label space $\mathcal{Y} = \{1, \dots, K\}$, consisting of K classes. Let X and Y be random variables that are distributed according to a joint distribution $\mathbb{P}_{X,Y}$ on $\mathcal{X} \times \mathcal{Y}$. A probabilistic model can be represented as $f : \mathcal{X} \to \mathbb{P}(\mathcal{Y})$, which is a map from the feature space to the space of probability distributions over the output space $\mathcal{Y}$. For multi-class classification problems with K classes, $\mathbb{P}(\mathcal{Y}) = \Delta_K$ (the $(K-1)$-dimensional probability simplex. Roughly speaking, a classifier is *calibrated* if its outputs coincide with probability distributions that match the empirical frequencies observed from realized outcomes. In multi-class classification, one distinguishes between different types of calibration. *Confidence calibration* (Niculescu-Mizil & Caruana, 2005; Guo et al., 2017; Kumar et al., 2018) only analyses the confidence score, i.e., the probability of the predicted class being calibrated. It is therefore the weakest notion of calibration. *Distribution calibration* (Bröcker, 2009; Vaicenavicius et al., 2019), which will be the focus in this paper, analyses whether all class probabilities are calibrated. Instead of only requiring calibrated marginal probabilities as in the definition of *classwise calibration* (Zadrozny & Elkan, 2002), it is defined via conditioning on the full probability vector; hence, it is the strongest notion of calibration.

Definition 1 A probabilistic multi-class classifier with output vector $f(X) \in \Delta_K$ is *calibrated in distribution* if for all $k \in \mathcal{Y} = \{1, \dots, K\}$ it holds that

$$\mathbb{P}(Y = k | f(X) = \boldsymbol{s}) = s_k,$$

with s_k being the k-th component of the probability vector $\boldsymbol{s} \in \Delta_K$.

To evaluate calibration, one typically makes use of calibration *errors*. In general, these errors measure the (average) discrepancy between the conditional probability distribution $\mathbb{P}_{Y|f(X)}$ and the predictions $f(X)$. Different calibration errors have been introduced to measure distribution calibration. One can in general differentiate between calibration errors defined as an *expectation* over a divergence $d : \Delta_K \times \Delta_K \to [0, \infty)$ between $f(X)$ and $\mathbb{P}_{Y|f(X)}$ (Popordanoska et al., 2022, 2024):

$$\mathrm{CE}_d(f) = \mathbb{E}\left[d\big(f(X), \mathbb{P}_{Y|f(X)}\big)\right], \tag{1}$$

and calibration errors defined via *integral probability metrics* (Müller, 1997), also called *kernel calibration errors* (Kumar et al., 2018; Widmann et al., 2019; Marx et al., 2024):

$$\mathrm{CE}_{\mathcal{H}}(f) = \sup_{\phi\in\mathcal{H}} \left|\mathbb{E}\left[\phi(f(X), Y)\right] - \mathbb{E}\left[\phi(f(X), Z)\right]\right|, \tag{2}$$

where $\mathcal{H}$ is typically chosen to be the unit ball of a reproducing kernel Hilbert space (RKHS) $\mathcal{H}$, and Z is a conditioning variable that is assumed to be calibrated.

For a given dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$, we denote $\widehat{\mathrm{CE}}(f) := \widehat{\mathrm{CE}}(f, \mathcal{D})$ as an estimator of CE based on the dataset $\mathcal{D}$.

A very common way to estimate the calibration error is to use binning (Zadrozny & Elkan, 2001; Naeini et al., 2015; Guo et al., 2017), i.e., partitioning the domain of the predictor variable $f(X)$ into a finite number of discrete intervals or *bins*, and estimating the calibration error by aggregating data within each bin. However, the piecewise nature of binning functions makes binned estimators non-differentiable and unsuitable for optimization tasks requiring gradient information, a property that we will need in Sect. 3. Binning also introduces a bias in estimating the conditional expectation $\mathbb{E}[Y|f(X)]$, because it replaces continuous variations with average values within bins. Furthermore, binned estimators usually measure a weaker form of calibration than distribution calibration as defined in Definition 1, namely classwise calibration, which only demands calibrated marginal probabilities. For these reasons, binning is not discussed in the following overview of existing estimators. Due to the nature of our problem, we make use of calibration estimators that are consistent, (asymptotically) unbiased and differentiable.

2.2 Overview of calibration errors and estimators

One of the first and still widely used metrics (Gupta et al., 2020; Murphy, 1973; Rahimi et al., 2020) that intrinsically also assesses calibration is the (expected) *Brier score* (Brier, 1950) . It is defined as

$$\mathrm{BS}(f) = \mathbb{E}\left[\left\|f(X) - \boldsymbol{e}_Y\right\|_2^2\right], \tag{3}$$

with $\boldsymbol{e}_Y$ being the one-hot-encoded vector of Y. Its estimator is given by the mean squared error between the predicted probability and actual outcome, averaged over all classes:

$$\widehat{\mathrm{BS}}(f) = \sum_{i=1}^{N}\sum_{j=1}^{K}(f_{ij} - \mathbb{1}_{(y_i=j)})^2, \tag{4}$$

where f_{ij} denotes the j-th entry of the vector $f(x_i)$. Being a proper scoring rule (Gneiting & Raftery, 2007), it does not only measure calibration, but can be decomposed into a calibration and a refinement loss term (Murphy, 1973; Kull & Flach, 2015), where the calibration loss consists of the expected L_2 error between the conditional distribution $P_{X|f(X)}$ and $f(X)$, hence Eq. (1) with d being the Euclidean distance. In their framework of *proper calibration errors*, Gruber and Buettner (2022) show that the Brier score also serves as an upper bound for other common calibration errors.

Popordanoska et al. (2022) proposed an estimator for the L^p *calibration error*, which for the case $p = 2$ directly corresponds to the calibration term in the decomposition of the Brier score:

$$\mathrm{CE}_p(f) = \left(\mathbb{E}\left[\left\|f(X) - \mathbb{P}_{Y|f(X)}\right\|_p^p\right]\right)^{\frac{1}{p}}. \tag{5}$$

They formulate an estimator of $\mathrm{CE}_p(f)$ using Beta and Dirichlet kernel density estimates for the binary and multi-class classification case, respectively. Precisely, they prove that

$$\widehat{\mathrm{CE}}_p(f) = \left(\frac{1}{n}\sum_{j=1}^{n}\left[\left\|\widehat{\mathbb{E}[y|f(x_j)]} - f(x_j)\right\|_p^p\right]\right)^{\frac{1}{p}} \tag{6}$$

is a point-wise consistent and asymptotically unbiased estimator, where the estimator of the conditional expectation is defined via kernel density estimation.

Using the Kullback–Leibler divergence D_{KL} as the divergence measure, one can define another proper calibration error (Popordanoska et al., 2024):

$$\mathrm{CE}_{KL}(f) := \mathbb{E}\Big[D_{KL}(\mathbb{E}[Y|f(X)], f(X))\Big], \tag{7}$$

which exactly forms the calibration error term of the popular log loss. An estimator is given by

$$\widehat{\mathrm{CE}}_{KL}(f) = \frac{1}{n}\sum_{j=1}^{n}\left\langle \widehat{\mathbb{E}[y|f(x_j)]}, \log\frac{\mathbb{E}[y|f(x_j)]}{f(x_j)}\right\rangle, \tag{8}$$

where the estimator $\widehat{\mathbb{E}[y|f(x_j)]}$ is again defined via kernel density estimation.

Widmann et al. (2019) introduced the *kernel calibration error* and multiple differentiable estimators for it. In its kernel-based formulation, it is defined using a matrix-valued kernel $k : \Delta_K \times \Delta_K \to \mathbb{R}^{K\times K}$ as follows:

$$\mathrm{CE}_k(f) = \left(\mathbb{E}\left[(\boldsymbol{e}_Y - f(X))^\top k(f(X), f(X'))(\boldsymbol{e}_{Y'} - f(X'))\right]\right)^{1/2}, \tag{9}$$

where (X', Y') is an independent copy of (X, Y) and $\boldsymbol{e}_Y$, $\boldsymbol{e}_{Y'}$ are the one-hot-encoded vectors of the random variables Y and Y'. We will make use of the computationally more feasible unbiased estimator they propose, defined as

$$\widehat{\mathrm{CE}}_k(f) = \frac{1}{\lfloor n/2\rfloor}\sum_{i=1}^{\lfloor n/2\rfloor} h_{2i-1,2i},$$

with k being a universal matrix-valued kernel and $h_{i,j}$ corresponds to the term in the expectation (9) evaluated on two instance-label pairs.

Marx et al. (2024) introduced a framework of calibration estimators for different types of calibration, which considers it as a distribution matching problem between (true) conditional distribution $\mathbb{P}_{Y|X}$ and the predicted distribution $\hat{\mathbb{P}}_{Y|X}$ induced by $f(X)$. Similar to Widmann et al. (2019), they proposed integral probability metrics to measure the distance between real conditional and predicted distribution. For $(X, Y) \sim \mathbb{P}$ and $(X, \hat{Y}) \sim \hat{\mathbb{P}}$ they define the calibration error as the Maximum-Mean-Discrepancy (MMD) between $\mathbb{P}$ and $\hat{\mathbb{P}}$:

$$\mathrm{CE}_{MMD}(f) = \sup_{\phi \in \mathcal{H}} \left| \mathbb{E}\left[\phi(Y, f(X))\right] - \mathbb{E}\left[\phi(\hat{Y}, f(X))\right] \right|, \tag{10}$$

where $\mathcal{H}$ is again the unit ball of an RKHS. For the classification case, they introduce a trainable calibration estimate for distribution calibration which measures the squared error as

$$\widehat{\mathrm{CE}}^2_{MMD} = \frac{1}{n(n-1)} \sum_{i=1}^{n} \sum_{j=1, j \neq i}^{n} h_{ij}, \tag{11}$$

where the terms h_{ij} are defined dependent on the type of calibration. They define z_i being a conditional random variable distributed according to $f(x_i)$, for $i \in \{1, \dots, N\}$ and

$$h_{ij} = k((y_i, z_i), (y_j, z_j)) + \sum_{y \in \mathcal{Y}} \sum_{y' \in \mathcal{Y}} q_i(y) q_j(y') k((y, z_i), (y', z_j)) - 2 \sum_{y \in \mathcal{Y}} q_i(y) k((y, z_i), (y_j, z_j)),$$

with $k : \Delta_K \times \Delta_K \rightarrow \mathbb{R}$ being a universal kernel function, and $q_i(y)$ the predicted probability for y given $f(x_i)$.

2.3 Statistical tests for calibration

As the estimators $\widehat{\mathrm{CE}}(f, \mathcal{D})$ reviewed in the previous section are computed from finite samples, their realised values inherit sampling variability, and hence comparing them in terms of their realized values does not take this randomness into account. Appendix A visualises the distribution of the estimators under the null hypothesis that they are calibrated; the dispersion shrinks with the sample size N, but even for large N randomness is non-negligible. Consequently a formal hypothesis test is needed when evaluating the calibration of a classifier $f : \mathcal{X} \rightarrow \Delta_K$ on a finite data set $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$:

$$H_0 : f \text{ is calibrated} \qquad H_1 : \neg H_0.$$

There are a number of established tests for calibration in classification problems. The classical Hosmer–Lemeshow test (Hosmer et al., 1997), initially developed as a goodness-of-fit test for the logistic regression model, considers a chi-squared distributed test statistic based on observed and expected frequencies of events. However, as it requires binning and can only be used to test for confidence calibration, we do not include it in this analysis. Widmann et al. (2019) derive a test for the kernel calibration error (CE_k), which takes the asymptotic normality of their proposed estimator into account. While it yields analytic p-values and is computationally light once the test statistic is calculated, its validity holds only for the specific kernel calibration estimator and its asymptotic normality.

Vaicenavicius et al. (2019) developed a general framework to evaluate calibration based on *consistency sampling* (Bröcker & Smith, 2007), a method that applies bootstrapping to estimate the distribution of the calibration estimator's values under the null hypothesis that f is calibrated. This is done by resampling new labels in each bootstrap iteration, based on the distribution that f predicts for the respective bootstrap sample. Given the (empirical) distribution function of the calibration estimator under the null hypothesis, one can then check how likely the given calibration estimate is under the assumption that f is calibrated. Being a nonparametric test, it can therefore be used to test for distribution calibration in combination with any of the calibration estimators mentioned above. For these reasons we adopt the nonparametric bootstrap test of Vaicenavicius et al. (2019) throughout the paper:

it is the only approach that (a) is valid for all considered calibration errors, (b) remains distribution-free, and (c) integrates seamlessly with our proposed algorithm.

3 Calibration of sets of probabilistic predictors

We will now come to the main part of this paper, namely the evaluation of calibration of not only one, but a set of classifier models. To this end, after introducing the necessary methodology, we introduce our proposed test in a step-wise manner.

3.1 Credal sets

Credal sets, generally defined as sets of probability distributions, form a way to represent disbelief or uncertainty about the *true* underlying probability distribution. Instead of committing to a single point prediction, using credal sets to represent uncertainty allows to stay imprecise, thereby avoiding prediction error if the uncertainty is too high. The sets can be obtained in various ways (Wang et al., 2025; Caprio et al., 2024), one direct way being via sets of probabilistic classifiers. In the following, let $\mathcal{F} := \{f^{(1)}, \dots, f^{(M)}\}$, with $f^{(i)} : \mathcal{X} \rightarrow \Delta_K$ the i-th probabilistic model in a set that contains M models in total. For each feature vector $x \in \mathcal{X}$, this yields a (credal) set of probability distributions $\mathcal{F}|_x = \{f^{(1)}(x), \dots, f^{(M)}(x)\} \subseteq \Delta_K$. A natural way to validate a representation of epistemic uncertainty through sets of predictors is to look at their possible *convex combinations*: If there is at least one prediction in the convex hull of $\mathcal{F}|_x$ that is calibrated, then one can argue that the set of predictors contains the ground truth aleatoric part of the uncertainty, hence fulfils a necessary requirement for representing epistemic uncertainty. Calibration here serves as a relevant necessary condition for validity; a calibrated combination indicates that the predictors are not systematically biased and that they can approximate the true data-generating process in a consistent manner. For each instance $x \in \mathcal{X}$, we now define the (credal) set $\mathcal{S}(\mathcal{F}, x)$ as the set of all possible convex combinations of predictors in $\mathcal{F}|_x$.

Definition 2 For a feature vector $x \in \mathcal{X}$, the credal set $\mathcal{S}(\mathcal{F}, x)$ is the set of all convex combinations of $\mathcal{F}|_x$:

$$\mathcal{S}(\mathcal{F}, x) = \left\{ f_\lambda(x) \in \mathcal{H} \,\middle|\, f_\lambda(x) = \sum_{i=1}^{M} \lambda_i(x) f^{(i)}(x) \,\middle|\, (\lambda_1, \dots, \lambda_M) \in \Delta_{M,\mathcal{X}} \right\},$$

where $\Delta_{M,\mathcal{X}} = \left\{ \lambda = (\lambda_1, \dots, \lambda_M) \middle| \, \lambda_i : \mathcal{X} \mapsto [0,1] \text{ and } \sum_{i=1}^{M} \lambda_i(x) = 1 \, \forall x \in \mathcal{X} \right\}$ denotes the set of all functions with co-domain being the $(M-1)$-simplex.

Here $\Delta_{M,\mathcal{X}}$ is a function space over $\mathcal{X}$, while in previous work (Mortier et al., 2023), Δ_M was the $(M-1)$-simplex of all (constant) convex combinations, that is, the same convex combination for all $x \in \mathcal{X}$ was analysed in order to test for calibration. Instance-dependent convex combinations offer the flexibility needed to capture varying degrees of aleatoric and epistemic uncertainty across the input domain. Different regions of the instance space may require different degrees of model blending to appropriately represent the level of uncertainty, particularly when some models are better calibrated than others. Def. 2 constructs credal sets via realizations of sets of predictors, which yield sets of probability distributions. From the perspective of imprecise probabilities, one might also define a *credal* predictor, which yields as output a set of probability distributions. The proposed framework also holds for this scenario, given the predictor outputs a *convex set* of extreme points.

3.2 Validity and calibration for sets of predictions

In the following, we formally define our notion of validity for set representations of epistemic uncertainty. It implies that for each $x \in \mathcal{X}$, the true underlying probability distribution $\mathbb{P}_{Y|X=x}$ is contained in the (credal) set of all possible convex combinations $\mathcal{S}(\mathcal{F}, x)$.

Definition 3 (*Validity of credal sets*) Let $\mathcal{F} = \{f^{(1)}, \dots, f^{(M)}\}$ be a set of predictors, with $f^{(i)} : \mathcal{X} \to \mathbb{P}(\mathcal{Y})$, and for each $x \in \mathcal{X}$, define the induced credal set $\mathcal{F}|_x$ as in Definition 2. Then $\mathcal{F}$ is *valid* if $\mathbb{P}_{Y|X=x} \in \mathcal{S}(\mathcal{F}, x)$ for all $x \in \mathcal{X}$.

In practice, testing validity requires having either access to the ground truth conditional distribution $\mathbb{P}_{Y|X}$ or a sufficient number of samples thereof, as assumed in Chau et al. (2024). Hence, in our setting, we make use of the *weaker* notion of calibration which results from conditioning on the predictions $f(X)$ instead of X. This way, we analyse whether there is a $\lambda \in \Delta_{M,\mathcal{X}}$ such that $\mathbb{P}_{Y|f_\lambda(x)} \in \mathcal{S}(\mathcal{F}, x)$. Similar as in Mortier et al. (2023), we now define a set of classifiers, or equivalently a credal classifier, as *calibrated* if there *exists* (at least) one calibrated convex combination of predictors. Definition 4 forms a generalization of Definition 4 of Mortier et al. (2023), where the coefficients of the convex combination do not form functions on the instance space, but are *constants*, i.e., $\lambda \in \Delta_M$.

Definition 4 Let $\mathcal{F} = \{f^{(1)}, \dots, f^{(M)}\}$ be a set of predictors, $f^{(i)} : \mathcal{X} \to \Delta_K$. We say that $\mathcal{F}$ is *calibrated in distribution* if there exists $\lambda \in \Delta_{M,\mathcal{X}}$ such that the *combined predictor* $f_\lambda : \mathcal{X} \to \mathcal{S}(\mathcal{F}, \mathcal{X})$ defined as

$$f_\lambda(x) = \sum_{i=1}^{M} \lambda_i(x) f^{(i)}(x)$$

is calibrated in distribution (Definition 1).

Note that calibration provides a necessary, but not *sufficient* condition for validity: A valid predictor is always calibrated, yet the reverse does not have to hold. In fact, there are (possibly) many calibrated predictors, as shown in Vaicenavicius et al. (2019). In Appendix B, we show that this is also the case for convex combinations of probabilistic models. However, if a predictor is *not* calibrated, one knows that it is also not valid.

3.3 A novel calibration test for sets of predictors

Our null and alternative hypothesis for testing the calibration of a set of predictors $\mathcal{F}$ can now be formulated as

$$H_0 : \quad \exists \lambda \in \Delta_{M,\mathcal{X}} \, \text{s.t.} f_\lambda \text{ is calibrated}, \quad H_1 : \neg H_0. \tag{12}$$

In order to analyse whether there is a calibrated convex combination, a natural approach is to analyse the one with the lowest calibration error. Hence, we define the *minimum calibration error* as follows:

$$\min_{\lambda \in \Delta_{M,\mathcal{X}}} g(f_\lambda), \tag{13}$$

where g is a calibration error.

For the experiments, we choose g to be equal to the errors proposed in Sect. 2, i.e., $g \in \{\mathrm{CE}_p, \mathrm{CE}_{KL}, \mathrm{CE}_{MMD}, \mathrm{CE}_k\}$. These have under suitable conditions the desirable property that f is calibrated if and only if $\mathrm{CE} = 0$, making them suitable for optimization. Furthermore, the respective estimators described in Sect. 2 are both consistent and at least asymptotically unbiased. Having an optimisation dataset $\mathcal{D}_{opt} = \{(x_i, y_i)\}_{i=1}^{\tilde{N}}$, one can represent the evaluations of the weight function $\lambda : \mathcal{X} \to \Delta_{M,\mathcal{X}}$ by an $\tilde{N} \times M$ matrix:

$$\Lambda = \begin{pmatrix} \lambda_1(x_1) & \dots & \lambda_M(x_1) \\ \vdots & \ddots & \vdots \\ \lambda_1(x_{\tilde{N}}) & \dots & \lambda_M(x_{\tilde{N}}) \end{pmatrix}. \tag{14}$$

Hence, for $\hat{g}$ being the (data-dependent) estimator of g, finding the *minimum empirical calibration error* can be formulated as follows:

$$\min_{\Lambda_{i,j},\, i \in 1,\dots,M, j \in 1,\dots,\tilde{N}} \hat{g}(f_\Lambda, \mathcal{D}_{val}) \tag{15}$$

with $f_\Lambda(x_i) = \sum_{m=1}^{M} \lambda_m(x_i) f^{(m)}(x_i) \in [0,1]^K$, having the constraint $\sum_{j=1}^{M} \Lambda_{ij} = 1$ for all $j \in \{1, \dots, \tilde{N}\}$.

As already described in Sect. 3.2, there are in general many calibrated functions f, implying the absence of a unique global minimum for the optimization problem (16). Therefore, in the optimisation, we make use of a combined loss function consisting of a proper scoring rule and the respective calibration estimator, weighted by a constant. Proper scoring rules intrinsically optimise for both a calibration error and an accuracy term and provide a stable optimisation. The specific optimisation method is described in Sect. 3.5.

3.4 Algorithmic procedure

Algorithm 1 Algorithm to test whether there exists a calibrated, instance-dependent convex combination of an ensemble of *M* classifier models.

```
1:  **Input:** validation set including features and labels $\mathcal{D}_{val} = \{(x_i, y_i)\}_{i=1}^{N}$, instance-
    wise evaluated point predictors $\mathcal{F}|_{(x_i)_{i=1}^N} = \left(f^{(1)}(x_i), \ldots, f^{(M)}(x_i)\right)_{i=1}^{N}$, estimator
    of calibration error $\hat{g}$, confidence level $\alpha$, number of bootstrapping iterations $D \in \mathbb{N}$
2:  **Output:** boolean stating whether to reject $H_0$, with $H_0$ as in (12)
3:  $\Lambda^* \leftarrow \underset{\Lambda_{i,j},\, i\in\{1,\ldots,M\}, j\in\{1,\ldots,N\}}{\arg\min} \hat{g}(f_\Lambda, \mathcal{D}_{val})$
4:  $f_{\Lambda^*} \leftarrow \Lambda^* \cdot \mathcal{F}$
5:  **for** $d = 1, \ldots, D$ **do**
6:      $\mathcal{D}_d \leftarrow \{\tilde{x}_1, \ldots, \tilde{x}_N\}$ bootstrap sample of instances
7:      **for** $n = 1, \ldots, N$ **do**
8:          sample $\tilde{y}_n \in \{1, \ldots, K\} \sim \mathrm{Cat}(f_{\Lambda^*}(x_n))$
9:      **end for**
10:     $\tilde{\mathcal{D}}_{val} \leftarrow \{(x_i, \tilde{y}_i)\}_{i=1}^{N}$            ▷ replace labels of validation set
11:     $t_{0,d} \leftarrow \hat{g}(f_{\Lambda^*}, \tilde{\mathcal{D}}_{val})$
12: **end for**
13: $q_{1-\alpha} \leftarrow (1-\alpha)$-quantile of empirical CDF $\hat{F}$ of $(t_{0,1}, \ldots, t_{0,D})$
14: $t \leftarrow \hat{g}(f_{\Lambda^*}, \mathcal{D}_{val})$
15: **if** $t > q_{1-\alpha}$ **then**
16:     reject $H_0$
17: **else**
18:     do not reject $H_0$
19: **end if**
```

We now explain in detail our new, adapted version of the statistical test proposed in Mortier et al. (2023), which in turn is based on the bootstrap test introduced by Vaicenavicius et al. (2019). Algorithm 1 shows the pseudo code of the test, which consists of the following steps:

First, using an appropriate optimization algorithm, the per-instance convex combination Λ^* with minimal calibration error is found (line 3). Using Λ^*, the predictions of the combined classifier f_{Λ^*} are computed (line 4). Third, the calibration test of Vaicenavicius et al. (2019) is employed to assess whether the predictions of f_{Λ^*} are calibrated (line 5–19). This statistical test is based on bootstrapping and *consistency resampling*, that is, resampling the labels from the distribution induced by f_{Λ^*}. This way, one is able to estimate the distribution of the calibration error estimator $\hat{g}$ under the null hypothesis that f_{Λ^*} is calibrated (see also Fig. 5 for an example).

For a given significance level α, the test rejects the null hypothesis if the value of the calibration error estimator is higher than the $(1-\alpha)$-quantile of the bootstrapped empirical distribution. In the other case, it cannot reject it. Algorithm 1 differs from the proposed algorithm in Mortier et al. (2023) in two important aspects:

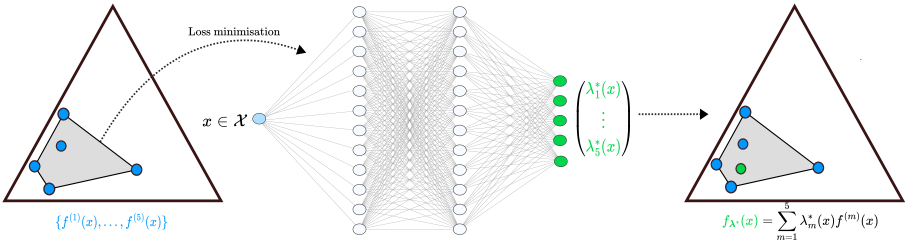

Loss minimisation
$x \in \mathcal{X}$
$\lambda_1^*(x)$
$\vdots$
$\lambda_5^*(x)$
$\{f^{(1)}(x), \dots, f^{(5)}(x)\}$
$f_{\lambda^*}(x) = \sum_{m=1}^{5} \lambda_m^*(x) f^{(m)}(x)$

Fig. 2 General concept of learning the optimal function $\lambda^* : \mathcal{X} \rightarrow \Delta_{M,\mathcal{X}}$, here for the example of $M = 5$ ensemble predictions. The neural network is trained using the combined calibration loss function as in Eq. (16), with the calibration estimators introduced in Sect. 2 for the calibration term. It predicts for a given instance value $x \in \mathcal{X}$ the optimal $\lambda^*(x)$ such that the empirical calibration error for the combined predictor f_{λ^*} is minimized

1. It allows for the case where the weights for the most calibrated convex combination *depend on the instance space*, i.e., for different regions in it, it accounts for the fact that some predictors might be better calibrated than others and vice versa.
2. It uses an *optimization* instead of a sampling-based approach: For constructing the distribution under the null hypothesis, the previous test used uniform sampling of the weights. Directly optimizing over the calibration error and performing the test on the found convex combination has two main advantages: (a), the uniform sampling of weights used in Mortier et al. (2023) does not lead to uniform sampling in the credal set, which we further elaborate on in Appendix D. By putting emphasis on combinations in the inner of the polytope, it could lead to a biased estimate of the distribution under the null hypothesis. Advanced sampling techniques that could result in a more uniform sampling in the credal sets, like rejection-sampling or triangulizations, are computationally costly. (b), the minimization step automatically "guides" the algorithm towards the region with low calibration errors, avoiding the need to explore the whole credal set. In Appendix C we show that using a universal approximator to learn the underlying convex combination leads to the test being asymptotically valid, i.e., controlling the Type 1 error. The optimisation step is explained in more detail in the following section.

3.5 A robust way of finding the most calibrated convex combination

We will now analyse further how optimization problem (15) can be solved in an efficient and robust manner. Specifically, we try to avoid the problem of *overfitting* on the empirical calibration error on the respective dataset: Since the latter is finite, there might be a mismatch between the estimated and the population-level calibration error. Classical gradient-based solvers, which purely optimise the calibration error on the same dataset that is used for the test, might therefore run into the problem of underestimating the true calibration error, thereby making the test more conservative. This can also be seen in the empirical results of Mortier et al. (2023).

Therefore, we suggest an alternative approach find the minimum in (15), which incorporates two important aspects:

- We use a neural network for learning the weight function $\lambda : \mathcal{X} \rightarrow \Delta_{M,\mathcal{X}}$, exemplary visualized in Fig. 2. This approach is similar to stacking (Wolpert, 1992), where a second meta-learner is trained to find the optimal combination of predictions. It is trained on a separate optimization dataset, and then used to predict the optimal convex combination for our test.

- As a loss function, we use the risk of a proper scoring rule ℓ, combined with a term controlling the calibration error, such that the true risk minimizer is given by

$$\lambda^* \in \underset{\lambda \in \Delta_{M,\mathcal{X}}}{\arg\min}\, \mathbb{E}\left[\ell(f_\lambda(x), y)\right] + \gamma \cdot g(f_\lambda). \tag{16}$$

We use $\hat{g} \in \{\widehat{\mathrm{CE}}_{KL}, \widehat{\mathrm{CE}}_2, \widehat{\mathrm{CE}}_k, \widehat{\mathrm{CE}}_{MMD}\}$ as estimates for the respective calibration error of interest, while $\gamma \geq 0$ denotes a weight factor. This approach takes recent insights (Popordanoska et al., 2022; Marx et al., 2024) that adding a calibrating penalty to a proper scoring rule can help ensuring calibration, into account, while avoiding trivial or degenerate solutions. In particular, many convex combinations can be calibrated (Appendix B), hence, the proper scoring rule serves as a *regularizer*, guiding the optimizer toward solutions that not only reduce calibration error but also maintain high predictive accuracy. Furthermore, we learn λ^* on a validation set but evaluate calibration on a separate test set. This split reduces the risk of overfitting to our chosen calibration metric, ensuring the resulting test does not become overly conservative.

4 Experimental results

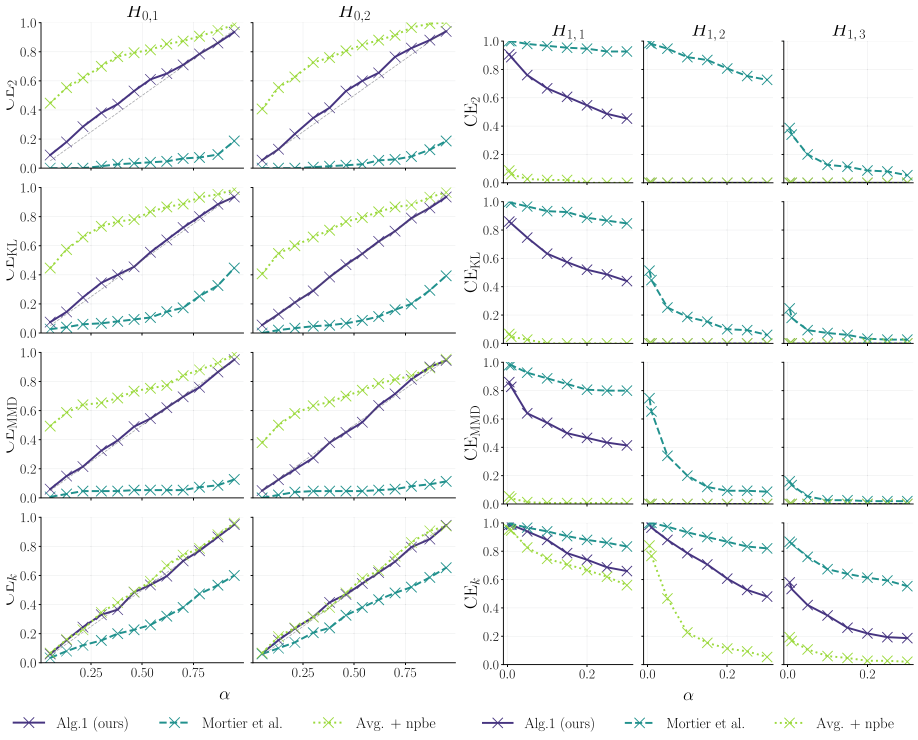
We evaluate our test both on synthetic and real-world data. A more detailed description of the experimental setup can also be found in Appendix E and the code for reproducing the experiments is available on GitHub.[1] For the experiments on synthetic data, a MLP architecture with 3 hidden layers and 16 neurons is used. Experimentally, we found that this simple architecture was sufficient to solve the optimization problem in Eq. (16) and learn the underlying weight function. For the experiments on real data, we use a more complex network architecture, which is described in Sect. 4.3. As a loss function, we use the combined loss as in (16), with the Brier score as a proper scoring rule, and $\gamma = 0.01$.

4.1 Binary classification

For illustration purposes, we first examine the case of binary classification by simulating $M = 2$ probabilistic predictors, each outputting a probability for the positive class. For each input x, the predictor's probabilities $f^{(1)}(x)$ and $f^{(2)}(x)$ are sampled from a Gaussian process (GP) constrained to [0, 1]. We then define a *true* calibrated predictor f^* that lies inside (under H_0) or outside (under H_1) the convex hull of the two predictors. Figure 9 shows the exemplary setting for this experiment. Under H_0, we consider two cases of non-instance and instance-dependence of $\lambda = (\lambda_1, \lambda_2)^T$:

1. $\boldsymbol{H_{0,1}}$: The calibrated predictor f^* is a *constant* convex combination of $f^{(1)}$ and $f^{(2)}$, where $\lambda_1^*(x) \equiv c \sim \mathit{Unif}([0,1])$, $\lambda_2^*(x) = 1 - \lambda_1^*(x)$.
2. $\boldsymbol{H_{0,2}}$: $\lambda^*(x)$ is a randomly generated polynomial function and $f^*(x) = \lambda^*(x) f^{(1)}(x) + (1 - \lambda^*(x)) f^{(2)}(x)$.

[1] https://github.com/mkjuergens/EnsembleCalibration.



(a) average Type 1 error

(b) average Type 2 error

Fig. 3 Average Type 1 and Type 2 error given a significance level (x-axis) for Algorithm 1 with $D = 100$ bootstrap iterations run on the binary classification experiment and the cases $\boldsymbol{H}_{0,1} - \boldsymbol{H}_{1,3}$ where the null hypothesis is true (**a**), and false (**b**), respectively. The average is taken over 200 resampling iterations. In each iteration we newly generate a dataset of size $N_{test} = 400$ on which the tests are performed. For the proposed test, we do the optimisation on a separate dataset $\mathcal{D}_{opt}$ of the same size. For comparison, the average Type 1 and Type 2 error of the previously proposed test (Mortier et al.) as well as the bootstrapping-based test of Vaicenavicius et al. applied to the mean prediction (Avg. + npbe) are shown

Under H_1, $f^*(x)$ lies strictly outside or near the boundary of the credal set. We generate 3 scenarios of increasing distance from it:

1. $\boldsymbol{H_{1,1}}$: $f^*(x)$ is set at a small ϵ-distance to one boundary.
2. $\boldsymbol{H_{1,2}}$: is sampled from a GP that remains close but outside the credal set.
3. $\boldsymbol{H_{1,3}}$: similar to $\boldsymbol{H_{1,2}}$, but allowing a larger maximum distance.

Using the described scenarios, Fig. 3 shows the Type 1 and Type 2 error of the proposed test, averaged over 200 runs of the experiment. For comparison, we include the results given by the algorithm of Mortier et al. (2023), as well as the algorithm of Vaicenavicius et al. (2019) applied to the mean of the ensemble, $f_{\lambda_{AVG}}(x) = \frac{1}{M}\sum_{i=1}^{M} f^{(i)}(x)$. The average Type 1 error of our testlies close to the chosen significance level. In some cases, it lies slightly above it, which is is due to the generalization error on the unseen data: Since we learn the optimally calibrated convex combination on a separate dataset, the learned convex combination on the test set might be not the one with lowest calibration error, due to the inherent randomness within the data. Nonetheless, with increasing sample size, we expect the Type 1 error to be controlled by the significance level (due to the test's asymptotic validity). The results show that the test of Mortier et al. is overly conservative, with a rejection rate far lower than the significance level. The test of Vaicenavicius applied to the mean prediction has a Type 1 error far higher than the significance level; which is expected, as the mean prediction might not lie close enough to the underlying conditional distribution for the test to not reject. Both our test and the test of Vaicenavicius applied to the mean show a high power for this setting, with almost zero Type 2 error also in the case where the true distribution lies in very close distance to the polytope. However, the conservative nature of the test of Mortier et al. leads to a higher Type 2 error, thereby revealing the provided benefit of our test in terms of power.

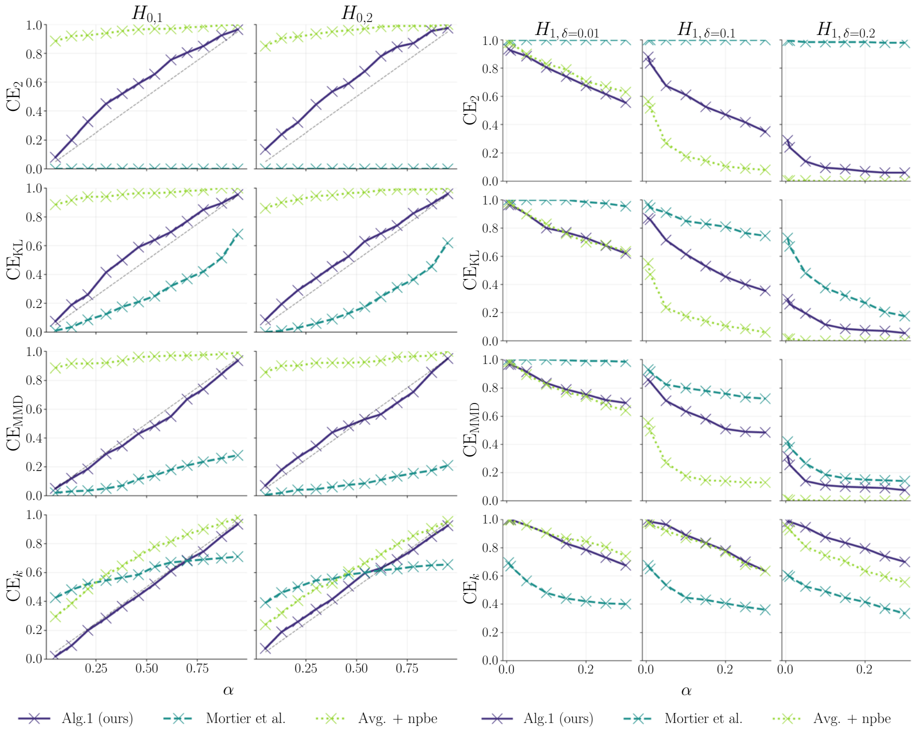
4.2 Multi-class classification

We consider $K > 2$ classes, and follow a Dirichlet-based scheme to generate an ensemble of M probabilistic predictors. Specifically, we generate a prior $\boldsymbol{p} \sim Dir(\mathbf{1})$ and predictions $f^{(1)}(x), \dots, f^{(M)}(x) \sim Dir(\frac{\boldsymbol{p}|x \cdot K}{u(x)})$, where $u : \mathcal{X} \rightarrow \mathbb{R}_{>0}$ is a function defining the epistemic uncertainty (i.e. the larger u, the higher is the *spread* between the predictions within the set) over the instance space. For the experiments, we use a constant $u(x) \equiv 0.5$. Under H_0, which we found to be a good tradeoff to evaluate Type 1 and 2 error of the test. We distinguish between

1. $\boldsymbol{H_{0,1}}$: $\lambda^*(x) \equiv \boldsymbol{c} \in [0, 1]$ where $\boldsymbol{c} \sim Dir(1, \dots, 1) \in \Delta_M$ (λ^* is constant across the instance space), and
2. $\boldsymbol{H_{0,2}}$: $\lambda^*(x) = (\lambda_1^*(x), \dots, \lambda_M^*(x)) \in \Delta_M$ with $\lambda_m^*(x) = \sum_{i=0}^{D} \beta_i x^i$ for $m = 1, \dots, M$ and $\sum_{m=1}^{M} \lambda_m^*(x) = 1$ (the components of λ^* form scaled polynomials of a certain degree).

For the alternative hypothesis, we select a random corner $f_c \notin \mathcal{F}$ of the simplex and then set the true underlying conditional distribution f^* as a point on the connecting line between the corner and the boundary point $f_b \in \mathcal{F}$ of the credal set: $f^*(x) = \delta \cdot f_c(x) + (1 - \delta) \cdot f_b(x)$. We define three cases with increasing distance to the credal set by varying the mixing coefficient δ: $\boldsymbol{H_{1,1}}$: $\delta = 0.01$, $\boldsymbol{H_{1,2}}$: $\delta = 0.1$ and $\boldsymbol{H_{1,3}}$: $\delta = 0.2$.

The resulting Type 1 and Type 2 errors of this experiment are shown in Fig. 4, again for our test and the two baselines. We see that the proposed test yields more reliable decisions, not heavily exceeding the given significance level while following it closely. The test of Mortier et al. is again overly conservative for all estimators, except for the kernel calibration error estimator $\widehat{\mathrm{CE}}_k$, which in general leads to unreliable results and low power. While the test of Vaicenavicius et al. applied to the mean prediction yields the highest power, it cannot be seen as a valid test, as it does not control the Type 1 error rate. Note here that we use the same datasets for all three tests, but we perform the optimisation for our proposed test on a separate dataset of the same size. The Type 2 error analysis shows that, except for the kernel calibration error, our proposed test has higher power than the previously proposed one, and it aligns better with the chosen significance level.



(a) average Type 1 error (b) average Type 2 error

Fig. 4 Average Type 1 and Type 2 error given a significance level (x-axis) for Algorithm 1 with $D = 100$ bootstrap iterations run on the multi-class classification experiment, and the settings $\boldsymbol{H}_{0,1} - \boldsymbol{H}_{1,3}$ where the null hypothesis is true (**a**), and false (**b**), resp. with $N_{test} = 400$ instances, $M = 10$ ensemble members, $K = 5$ classes and uncertainty $u = 0.5$. The average is taken over 200 resampling iterations. For comparison, the average Type 1 and Type 2 error of the previously proposed test (Mortier et al.) as well as the bootstrapping test applied to the mean prediction (Avg. + npbe) are shown

4.3 Experiments on real-world data

Since the true data-generating distribution is unknown in real-world datasets, we cannot directly quantify Type 1 or Type 2 errors. Instead, we demonstrate the practical usefulness of our test by applying it to the two standard image classification tasks CIFAR-10 and CIFAR-100 (Krizhevsky et al., 2009). More specifically, we apply our test on the predictions of three ensemble methods, combined with two different architectures that are trained on the two datasets, respectively. We train a deep ensemble (DE, (Lakshminarayanan et al., 2017)), a dropout network with dropout rate 0.5 (DN(0.5), (Gal & Ghahramani, 2016)) and a dropout network with dropout rate 0.2 (DN(0.2)). For the deep ensemble, we train 10 different models using different weight initializations, while in the dropout networks, dropout is applied after specific layers. A higher dropout rate injects more parameter noise during training, increasing predictor diversity and thus enlarging the credal set, which is why we analyse the two different cases. After training, we obtain a set of different predictions from these models, which we then use to run Algorithm 1. As model architectures, we use two architectures of different model complexity: the ResNet-18 (He et al., 2016) (11.6×10^6 parameters) and the VGG-19 (Simonyan & Zisserman, 2014) (138×10^6 parameters). For the optimisation part, we leverage the same architecture, i.e. a pre-trained ResNet-18 or VGG-19, to embed images and attach a fully-connected layer with 32 neurons to predict the weight function λ. Section E describes the full experimental setup. The results are shown in Table 1. For each model and calibration estimator, we run the respective bootstrapping test (line 5–19 of Algorithm 1), with the optimal weight function λ^* learned by the neural network. Similarly as in the synthetic experiments, we also perform the bootstrapping part of the test with the mean predictor (Algorithm 1 from line 5 with $f_{\lambda_{AVG.}} = \frac{1}{M}\sum_{j=1}^{M} f^{(j)}$). The results differ significantly, depending on the used calibration error. In general, using the calibration error CE_{KL} led to the fewest rejections (i.e. the highest p values), potentially

Table 1 Calibration test results on CIFAR-10 and CIFAR-100 for the significance level $\alpha = 0.05$ and $D = 100$ bootstrap iterations

DATA	ARCH	TYPE	ERR	λ_{AVG}				λ^*			
				Acc	H_0	p val	$\hat{g}$	Acc	H_0	p val	$\hat{g}$
CIFAR-10	ResNet-18	DE	CE_{KL}	0.869	¬REJ	1.000	0.259	0.873	¬REJ	0.990	0.262
			CE_2	0.869	¬REJ	0.970	0.125	0.872	¬REJ	0.950	0.121
		DN(0.2)	CE_{KL}	0.847	¬REJ	0.810	0.293	0.848	¬REJ	0.780	0.297
			CE_2	0.847	¬REJ	0.800	0.129	0.849	¬REJ	0.760	0.295
		DN(0.5)	CE_{KL}	0.507	¬REJ	0.800	0.330	0.507	¬REJ	0.780	0.329
			CE_2	0.507	¬REJ	0.220	0.104	0.507	¬REJ	0.350	0.106
	VGG-19	DE	CE_{KL}	0.900	¬REJ	0.530	0.176	0.902	¬REJ	0.420	0.166
			CE_2	0.900	¬REJ	0.150	0.079	0.901	¬REJ	0.180	0.084
		DN(0.2)	CE_{KL}	0.903	REJ	0.000	0.134	0.903	REJ	0.000	0.133
			CE_2	0.903	REJ	0.000	0.042	0.903	REJ	0.000	0.042
		DN(0.5)	CE_{KL}	0.899	REJ	0.000	0.131	0.896	REJ	0.000	0.129
			CE_2	0.899	REJ	0.010	0.047	0.896	REJ	0.000	0.056
CIFAR-100	ResNet-18	DE	CE_{KL}	0.586	¬REJ	1.000	1.200	0.589	¬REJ	1.000	1.203
			CE_2	0.586	REJ	0.000	0.476	0.589	REJ	0.000	0.482
		DN(0.2)	CE_{KL}	0.264	¬REJ	1.000	2.123	0.265	¬REJ	1.000	2.126
			CE_2	0.264	REJ	0.000	0.552	0.265	REJ	0.000	0.551
		DN(0.5)	CE_{KL}	0.352	¬REJ	1.000	2.123	0.353	¬REJ	1.000	2.136
			CE_2	0.352	REJ	0.000	0.620	0.351	REJ	0.030	0.624
	VGG-19	DE	CE_{KL}	0.647	¬REJ	0.510	0.925	0.648	¬REJ	0.390	0.927
			CE_2	0.647	REJ	0.000	0.347	0.647	REJ	0.000	0.346
		DN(0.2)	CE_{KL}	0.621	REJ	0.030	0.999	0.620	REJ	0.000	1.006
			CE_2	0.621	REJ	0.020	0.343	0.620	REJ	0.020	0.343
		DN(0.5)	CE_{KL}	0.588	¬REJ	0.210	1.282	0.587	¬REJ	0.410	1.274
			CE_2	0.588	REJ	0.010	0.417	0.587	REJ	0.010	0.416

For each architecture (ResNet-18 or VGG-19) and model type (Deep Ensemble (DE) or MC Dropout (DN(p)) with rate $p \in \{0.2, 0.5\}$, we show results of running the proposed test using two different calibration errors, $g \in \{\mathrm{CE}_{KL}, \mathrm{CE}_2\}$. For $g = \mathrm{CE}_{KL}$, log loss, for $g = \mathrm{CE}_2$ the Brier score is used as the proper scoring rule for optimization. Columns 4–7 show accuracy, test decision, p value, and value of the test statistic for the average ensemble combination (λ_{AVG}), and columns 8–11 for the learned convex combination (λ^*)

because the models were trained with a log-loss objective as proper scoring rule, which aligns more closely with KL-based calibration measures. In general, for the CIFAR-100 dataset with more classes, the test rejects more often, and the dropout networks are less calibrated. In most cases, the learned convex combination λ^* leads to a slight increase in accuracy for the combined prediction f_{λ^*}.

5 Discussion

We developed a novel statistical test for the calibration of epistemic uncertainty representations based on sets of probability distributions. It is defined on an instance-based level, thereby making the test more flexible. As calibration forms a necessary condition for *validity*, we claim that by testing for it, one can safely detect scenarios where the credal set is *not* valid. For this case, the next step to include would be *actionability*, i.e., ways to increase the size of the credal set to include at least one calibrated convex combination. Here, we model calibration as a property that applies across the entire instance space. Alternatively, calibration could be viewed as an instance-specific concept, allowing analysis in different regions of the instance space. However, there has been limited research on this form of calibration, sometimes referred to as conditional or *local* calibration (Luo et al., 2022), and the challenge of partitioning the instance space in a way that enables the computation of expectations remains unresolved. Interesting future work includes the application of the test to other credal set representations such as interval-neural networks (Wang et al., 2025), credal sets based on relative log-likelihood (Löhr et al., 2025) and bootstrapped ensembles, as well as the analysis of the regression case.

Appendix A Estimator analysis

In this section we empirically analyse the distribution of the chosen calibration estimators and their behaviour with respect to the distance from the underlying ground truth probability distribution. Figure 5 shows the distribution of the estimators under the null hypothesis. In the figure, we see that the mean of the estimators $\widehat{\mathrm{CE}}_2$ and $\widehat{\mathrm{CE}}_{KL}$ is slightly above zero—since they are only asymptotically unbiased—while $\widehat{\mathrm{CE}}_{MMD}$ and $\widehat{\mathrm{CE}}_k$ also empirically show their unbiasedness. The kernel calibration estimator $\widehat{CE}_k$ is symmetrically distributed around zero—its asymptotic normality was also shown by Widmann et al. (2019). As the estimators $\widehat{CE}_k$ and $\widehat{CE}_{MMD}$ can obtain negative values, we use their squared versions for the optimization in Eq. (16).

Figure 6 shows the values of different calibration estimators in dependence of the position in the simplex, where the true underlying calibrated convex combination f_{λ^*} is set with $\lambda^* = (0.1, 0.1, 0.8)$, and the predictions are sampled from a Dirichlet distribution. We see that by optimizing over λ, the true calibrated convex combination is learned differently "well enough". However, λ^* still lies within a certain region of low estimator values. For $\widehat{CE}_k$, we see in general more noisy behaviour in the region around the ground truth distribution (red dot).

We also do a similar analysis in λ-space: In Fig. 7, we show the value of the calibration estimators not in distance to the ground truth probability distribution, but the ground truth weight vector $\lambda \in \Delta_3$, given the predictions of two predictors $\{f^{(1)}(x), f^{(2)}(x), f^{(3)}(x)\}$ and a ground truth probability distribution $f_{\lambda^*}(x) = 0.1 \cdot f^{(1)}(x) + 0.1 \cdot f^{(2)}(x) + 0.8 \cdot f^{(3)}(x)$, that

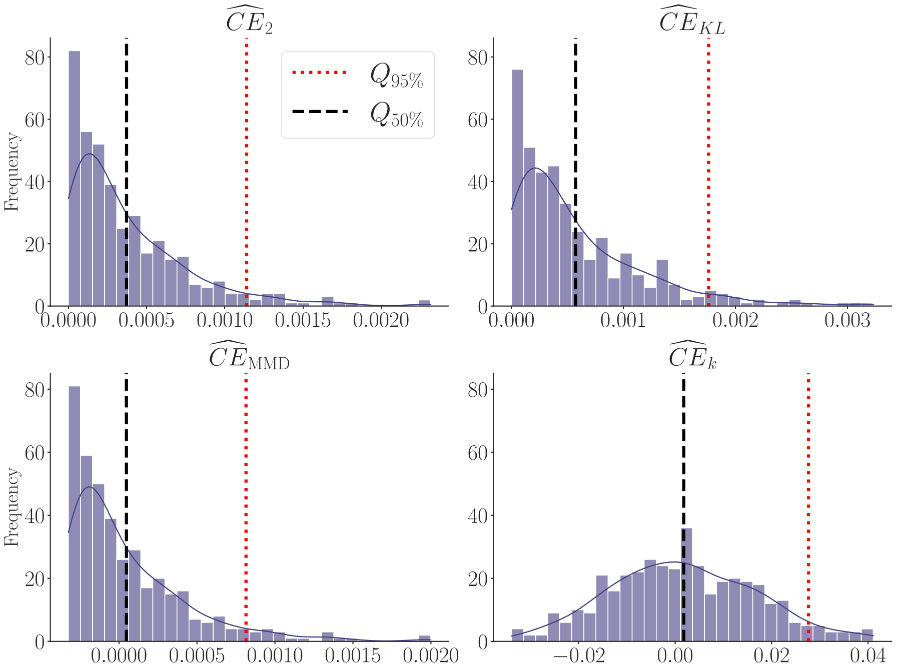


Fig. 5 Distributions of calibration error estimators under the null hypothesis. Histograms (with kernel density estimates) of the estimated calibration error $g \in \{\widehat{CE}_2, \widehat{CE}_{KL}, \widehat{CE}_{\text{MMD}}, \widehat{CE}_k\}$ are shown. Predicted probabilities were sampled from a uniform Dirichlet distribution over three classes ($K = 3$) and repeated $N = 1000$ times to simulate a perfectly calibrated model. Labels were sampled from the corresponding categorical distribution in $D = 500$ resampling steps. Vertical dashed black lines indicate the mean values of each statistic. Vertical dashed red lines indicate the 95% quantile of the empirical distribution

is $\lambda^* = (0.1, 0, 1, 0.8)^T$. We see that here the heatmaps vary more significantly for the different calibration estimators.

Appendix B Non-uniqueness of calibrated models

Using simple examples, we will show the non-uniqueness of calibrated convex combinations, that is, that there are often many different (possibly instance-dependent) convex combinations leading to a calibrated predictor. Using a proposition proven by Vaicenavicius et al. (2019), which states that conditioning Y on any measurable function h yields a calibrated predictor, we show this with a concrete example.

Example 1 (*Many calibrated classifiers*) For the following, assume the case of having 3 classes, i.e. $\mathcal{Y} = \{1, 2, 3\}$. We have (Vaicenavicius et al. 2019, Proposition 1) for any measurable function $h : \mathcal{X} \mapsto \mathcal{Z}$ with $\mathcal{Z}$ being some measurable space, that the function

$$f(X) := \mathbb{P}[Y \in \cdot | h(X)]$$

is a calibrated classifier. As an example, taking a constant $h(x) = c \in \mathbb{R}\ \forall x \in \mathcal{X}$, the classifier that simply predicts the marginal class probabilities

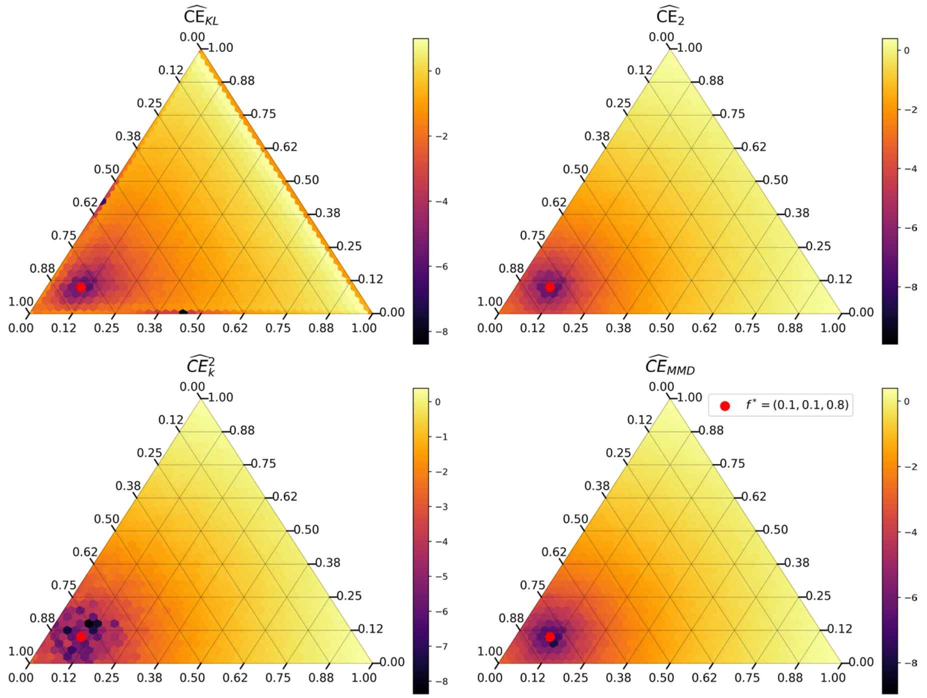


Fig. 6 Simple experiment that illustrates the behaviour of the calibration estimators for the case of a *non-instance-dependent* underlying ground truth distribution. Log-transformed values of the empirical calibration error $\hat{g}(\hat{f}, \mathcal{D})$ where $\hat{g} \in \{\widehat{\text{CE}}_{KL}, \widehat{\text{CE}}_2, \widehat{\text{CE}}_k^2, \widehat{\text{CE}}_{MMD}\}$, in dependence of the (constant) predictor $\hat{f}$ within the probability simplex, for the case of $K = 3$ classes. $N = 2000$ labels y_i in $\mathcal{D}$ were generated from the ground truth categorical distribution, $y_i \sim \text{Cat}(f^*)$, $i = 1, \ldots, 2000$. Here, we set $f^*(x) \equiv (0.1, 0.1, 0.8)^T$ constant for all $x \in \mathcal{X}$. The red point corresponds to f^* and represents the theoretical minimum of the calibration error

$$f \equiv \begin{pmatrix} \mathbb{P}(Y = 1) \\ \mathbb{P}(Y = 2) \\ \mathbb{P}(Y = 3) \end{pmatrix}$$

is calibrated for $\mathcal{Y} = \{1, 2, 3\}$. Further, if we take $h : \mathcal{X} \mapsto \mathcal{X}$ with $h(x) = x$, then

$$f' \equiv \begin{pmatrix} \mathbb{P}(Y = 1|X) \\ \mathbb{P}(Y = 2|X) \\ \mathbb{P}(Y = 3|X) \end{pmatrix}$$

is also a calibrated classifier. To make this example more concrete, let $\mathcal{X} = \{1, 2, 3\}$ with $\mathbb{P}(X = x) = \frac{1}{3} \forall x \in \mathcal{X}$. Further let again $\mathcal{Y} = \{1, 2, 3\}$ and

$$\mathbb{P}(Y = i|X = j) = \begin{cases} 1, & i = j \\ 0, & \text{else} \end{cases}$$

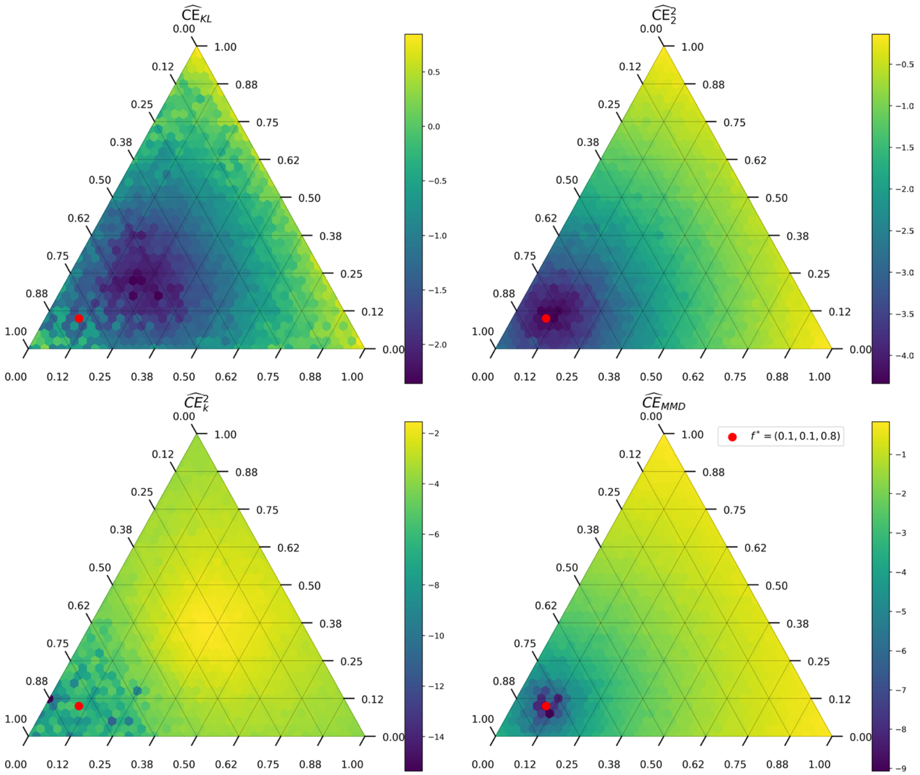


Fig. 7 Heatmaps of the values of the chosen calibration estimators for a given location within the space of weights $\lambda \in \Delta_3$. The predictions $f^{(i)} \in \{f^{(1)}, f^{(2)}, f^{(3)}\}$ are generated according to a Dirichlet distribution with parameters $\alpha_j = 1 + \mathbb{1}_{i=j} \cdot 10$ for $j \in \{1, 2, 3\}$

for $i, j \in \{1, 2, 3\}$. Then $\mathbb{P}(Y = i) = \sum_{j=1}^{3} \mathbb{P}(Y = i|X = j)\mathbb{P}(X = j) = \frac{1}{3}$ for $i \in \{1, 2, 3\}$, and

$$f(x) \equiv \begin{pmatrix} 1/3 \\ 1/3 \\ 1/3 \end{pmatrix}, \quad f'(x) = \mathbf{e}_x$$

where $\mathbf{e}_x$ denotes the x-th unit vector, are both calibrated predictors.

Going from the case of one single classifier, similarly, one can also show that there usually exists *various calibrated convex combinations* for a set of classifiers. The following example illustrates this for the simple case of binary classification.

Example 2 Let us look at a very simple problem of binary classification with $\mathcal{Y} = \{1, 2\}$ and $\mathcal{X} = \{-1, 1\}$. Assume

$$\mathbb{P}(Y = i|X = x_j) = \begin{cases} 1, & (i, x_j) \in \{(1, -1), (2, 1)\} \\ 0, & \text{else,} \end{cases}$$

and $\mathbb{P}(X = x) = \frac{1}{2}, \quad x \in \mathcal{X}$. Then the two classifiers

$$\begin{aligned} f_1 &: \mathcal{X} \rightarrow \Delta^2 \\ &x \mapsto \begin{pmatrix} \frac{1}{2} \\ \frac{1}{2} \end{pmatrix}, \quad x \in \{-1, 1\} \end{aligned}$$

and

$$\begin{aligned} f_2 &: \mathcal{X} \rightarrow \Delta^2 \\ &x \mapsto \begin{cases} \begin{pmatrix} 1 \\ 0 \end{pmatrix}, & x = -1 \\ \begin{pmatrix} 0 \\ 1 \end{pmatrix}, & x = 1 \end{cases} \end{aligned}$$

are both calibrated (which can easily be seen by Proposition 1 of Vaicenavicius et al. (2019) and conditioning on $h_1(x) = c$ and $h_2(x) = x$). Further, they can be written as convex combinations of the two classifiers $m_1(x) = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \forall x$ and $m_2(x) = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \forall x$ using the two different convex combinations

$$\begin{aligned} \lambda_1 &: \mathcal{X} \rightarrow \Delta^2 \\ &x \mapsto \begin{cases} \begin{pmatrix} 0 \\ 1 \end{pmatrix}, x = -1 \\ \begin{pmatrix} 1 \\ 0 \end{pmatrix}, x = 1 \end{cases} \end{aligned}$$

and

$$\begin{aligned} \lambda_2 &: \mathcal{X} \rightarrow \Delta^2 \\ &x \mapsto \begin{pmatrix} \frac{1}{2} \\ \frac{1}{2} \end{pmatrix}, x \in \{-1, 1\}. \end{aligned}$$

Appendix C Validity of the proposed algorithm

In the following, we prove the asymptotic validity of the derived test.

Theorem 1 (Asymptotic validity of Algorithm 1) Let $\mathcal{D}_{opt} = \{(x_i, y_i)\}_{i=1}^n$ and $\mathcal{D}_{val} = \{(x'_i, y'_i)\}_{i=1}^m$ be i.i.d. data generated from the underlying distribution $\mathbb{P}_{X,Y}$. Denote by

$$\hat{\lambda}_n \in \arg\min_{\Lambda} \frac{1}{n} \sum_{i=1}^{n} \ell(f_\Lambda(x_i), y_i) + \gamma\, \widehat{g}_n(f_\Lambda, \mathcal{D}_{opt})$$

the weight function that is the risk minimizer of the empirical version of Eq. (16) evaluated at $\mathcal{D}_{opt}$, where ℓ is the (strictly) proper Brier score or log loss and $\widehat{g}_n$ is a consistent, differentiable and (asymptotically) unbiased estimator of one of the calibration errors in Sect. 2.

Define $t_{m,n} := \hat{g}_m(f_{\hat{\lambda}_n}, \mathcal{D}_{val})$ the test statistic evaluated on the combined predictor, and let $c_{m,D,\alpha} = \hat{F}^{-1}_{m,D}(1-\alpha)$ be the $(1-\alpha)$-quantile of the empirical cumulative distribution function $\hat{F}_{m,D}$ of the bootstrap replicates $t_{0,1}, \dots, t_{0,D}$ generated by the consistency–resampling scheme of Vaicenavicius et al. (2019) (line $5-19$ of Algorithm 1). Furthermore, assume that the set of predictors $\mathcal{F} = \{f^{(1)}, \dots, f^{(M)}\}$ is such that the mapping $\lambda \mapsto f_\lambda$ is continuous over $\mathcal{X}$ and for the respective calibration function g all regularity conditions are fulfilled such that f is calibrated if and only if $g = 0$. Then Algorithm 1 rejects H_0 whenever $t_{m,n} \geq c_{m,D,\alpha}$. Then, under H_0,

$$\lim_{\substack{m,n \to \infty \\ D \to \infty}} \mathbb{P}(t_{m,n} \geq c_{m,D,\alpha} | H_0) = \alpha.$$

Proof In the following, we assume that the null hypothesis is true, i.e., there exists (at least) one underlying function λ such that the combined predictor f_λ is calibrated.

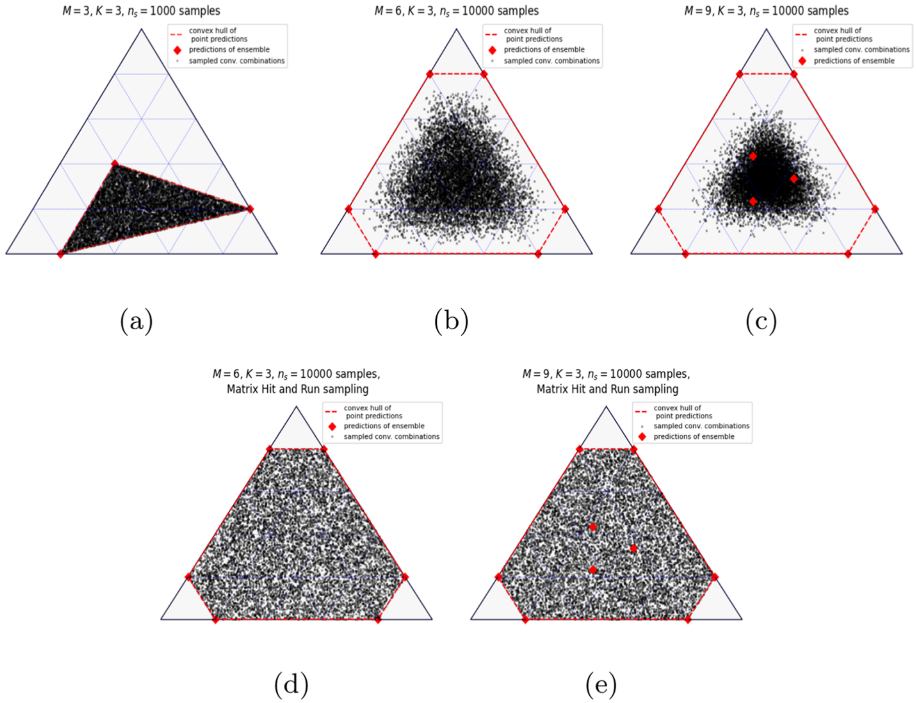
1. Consistency: Because ℓ is strictly proper, the population level risk in Eq. (16) is uniquely minimised by some $\lambda^\star \in \Delta_{M,\mathcal{X}}$ with respective combined predictor $f_{\lambda^\star}$. As the expectation of ℓ can be decomposed into a calibration and refinement term, and $g(f) = 0$ for any calibrated predictor f, we have that $g(f_{\lambda^\star}) = 0$.

   By the arg-min consistency theorem (Thm. 5.7 of van der Vaart (2000)) it holds that the empirical risk minimizer converges to the population level one, i.e. $\|\hat{\lambda}_n - \lambda^\star\|_{L_1} \to 0$ as $n \to \infty$, and hence we have $\|f_{\hat{\lambda}_n} - f_{\lambda^*}\|_{L_1} \to 0$.
2. Asymptotic distribution: Under the null hypothesis, for each of the calibration errors we have $g(f_{\lambda^*}) = 0$. Further, taking the asymptotic behavior of U-statistics (van der Vaart, 2000) into account, we have $\sqrt{m}\,\hat{g}_m(f_{\lambda^*}, \mathcal{D}_{val}) \to Z$, $m \to \infty$, for some random variable with limiting distribution F_Z. By Rubin's theorem (see Theorem 7.9 in Dümbgen (2017)) it holds that $\sqrt{m}\, t_{m,n} \to Z$, $m, n \to \infty$.
3. Bootstrap validity: For sake of simplicity, set $D = m$. Under H_0, the consistency-bootstrap (line 5 to 19 of Algorithm 1, Vaicenavicius et al. (2019)) guarantees that conditionally on the validation data set it holds that $\sqrt{m}\, c_{m,D,\alpha} \to F_Z^{-1}(1-\alpha)$. This is because $\sup_t |\hat{F}^{-1}_{m,D}(t) - F^{-1}_{t_{m,n}}(t)| \to 0$ (compare to Section 23.2.1 in van der Vaart (2000)).
4. Control of Type 1 error: The convergence of the bootstrap quantile against the quantile of the limiting distribution implies $\mathbb{I}\{t_{m,n} \geq c_{m,D,\alpha}\} \to \mathbb{I}\{Z \geq F_Z^{-1}(1-\alpha)\}$. Taking expectations yields: $\mathbb{P}(\text{reject}|H_0) = \mathbb{P}(t_{m,n} \geq c_{m.B.\alpha}) \to 1 - F_Z(F_Z^{-1}(1-\alpha)) = \alpha$.

□

Appendix D Uniform sampling in the credal set

In the work of Mortier et al. (2023), a sampling within the polytope of convex combinations of predictions is done by sampling weights $\lambda \in \Delta_M \sim Dir(1, \dots, 1)$. This however does in general not lead to a uniform sampling within the polytope, as is exemplary shown in Fig. 8. As soon as we have more than $M = 3$ predictors, the sampled predictions start "accumulating" in the centre of the simplex, due to the fact that there are more possible combinations in the centre than for the regions in the boundary. Possible ways to achieve uniform sampling is by using triangulization methods, rejection sampling or Markov Chain



(a) (b) (c)

(d) (e)

Fig. 8 Sampling example for $K = 3$: The point predictions $f^{(i)}(x)$ are visualized as red dots on the corners and/or the inside of the 2-simplex. In (**a**) and (**b**), and (**c**) samples $f_{\lambda_1}, \dots, f_{\lambda_n}$ are generated by sampling the weights of the convex combination from the uniform Dirichlet distribution, $\lambda_i \sim Dir(1, \dots, M)$. In (**d**) and (**e**), an MCMC approach is used to uniformly sample in the polytope of convex combinations

Monte Carlo sampling methods, however, these methods come with increased computational effort.

Appendix E Experimental setup

In this section we explain the experimental setup for the experiments conducted in Sect. 4.

E.1 Synthetic data

For optimising the weights λ in the synthetic experiments, we use an MLP with 3 hidden layers each consisting of 16 neurons that is trained on an optimization dataset of size $N_{opt} = 400$. Both tests (the previous one proposed by Mortier et al. and ours) are performed on a validation set set of size $N_{val} = 400$ (Fig. 9).

For the case of **binary classification**, we draw $\{x_i\}_{i=1}^{N} \sim U([0, 5])$, and generate predictions from two probabilistic predictors $\{f^{(1)}(x), f^{(2)}(x)\}$ by sampling from a Gaussian process with an rbf kernel, whose outputs are constrained to output probabilities [0, 1], using min-max sclaing. Under H_0, we generate the probability for class 1, $f^*(x) = \lambda^* f^{(1)}(x) + (1 - \lambda^*) f^{(2)}(x)$ with λ^* generated as

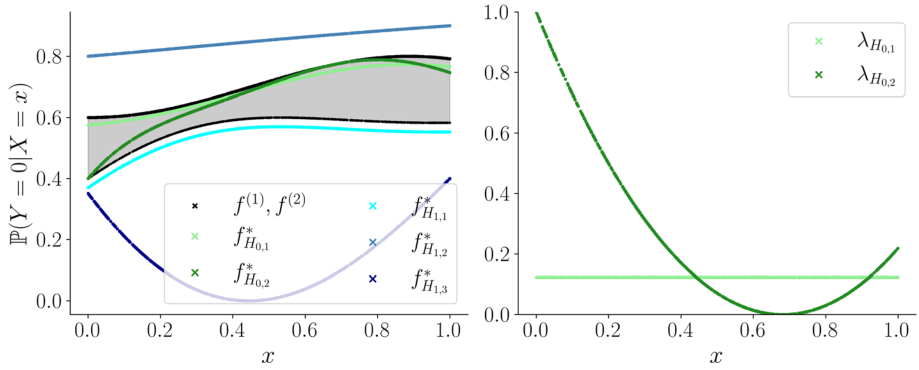


Fig. 9 1-simplex spanned by the ensemble members $\{f^{(1)},f^{(2)}\}$ (grey) that are generated according to the binary classification setting as described in Sect. 4.1, and the underlying ground truth calibrated predictions for test cases $H_{0,1}$ - $H_{1,3}$ described in Sect. 4.1 (left). In the right graphic, the respective weights of the underlying convex combination for the cases $H_{0,1}$ (constant) and $H_{0,2}$ (instant-dependent polynomial) are visualised as a function of the instance space

- $\boldsymbol{H_{0,1}}$: λ^* is a constant and sampled as $\lambda^* \sim U(0,1)$,
- $\boldsymbol{H_{0,2}}$: $\lambda^* : \mathcal{X} \rightarrow [0,1]$ is a polynomial of degree D. In the experiments, we set $D = 2$.

Under H_1, $f^*(x)$ lies outside the credal set with

- $\boldsymbol{H_{1,1}}$: $f^*(x)$ lies at an ϵ-distance to one of the boundaries of the credal set. We sample $\epsilon \sim U([0, 0.02])$.
- $\boldsymbol{H_{1,2} - H_{1,3}}$ reflect increasing distance to the boundary by sampling a new GP that intentionally remains outside the credal set.

For the **multi-class classification** case, we again draw $\{x_i\}_{i=1}^N \sim U([0,5])$ and generate the set of probabilistic predictors $f^{(1)}, \dots ,f^{(M)}\}$ as follows: For each x_i, draw a prior $\boldsymbol{p} \in \mathbb{R}^K \sim Dir(1, \dots, 1)$, then sample $f^{(m)}(x_i) \sim Dir(\frac{\boldsymbol{p} \cdot K}{0.5})$. Under H_0, the weight function $\boldsymbol{\lambda}^*$ is generated as

- $\boldsymbol{H_{0,1}}$: $\boldsymbol{\lambda}^* \sim Dir(1, \dots, 1)$
- $\boldsymbol{H_{0,2}}$: $\boldsymbol{\lambda}^* = (\lambda_1^*, \dots, \lambda_M^*)$ with $\lambda_m^* : \mathcal{X} \rightarrow [0,1]$; the components of $\boldsymbol{\lambda}^*$ are randomly generated and scaled polynomials of degree $D = 2$.

Under H_1, $f^c(x)$ is randomly chosen as one of the corners of the simplex, outside of the convex set. $f^*(x)$ is then given by the point prediction on the line segment between $f^c(x)$ and the boundary point $f^b(x)$ in the credal set, $f^*(x)(x) = \delta f^c(x) + (1-\delta) f^b(x)$, with

- $\boldsymbol{H_{0,1}}$: $\delta = 0.01$,
- $\boldsymbol{H_{0,2}}$: $\delta = 0.1$,
- $\boldsymbol{H_{0,3}}$: $\delta = 0.2$.

In both settings, the labels $\{(y_i)\}_{i=1}^N$ are then sampled from the resulting categorical distribution with parameter $f^*(x_i)$.

E.2 Real data

Here we provide details on the data preprocessing, model architectures, hyper-parameters, and calibration-test implementation used in the real-world experiments.

Data splitting: We split the data into training set, (used to train the ensembles), validation and test set. The validation set together with the predictions of the models are then used to optimise for λ^*, and the test is performed on the test data.

- CIFAR-10: 50000 training, 5000 validation and 5000 test samples,
- CIFAR-100: 60000 training, 5000 validation and 5000 test samples.

Models: We use the following training parameters to train the deep ensemble and dropout models, in order to obtain the credal set of predictions:

- adam optimizer with lr = 0.001,
- loss function: cross-entropy (log loss)
- batch size: 128
- number of epochs: 50
- early stopping: monitor validation loss, stop if does not decrease for patience = 10 epochs.

Optimization of weights: We concatenate a small MLP consisting of 3 hidden layers and 32 neurons to the same VGG-19/ResNet-18 architecture that has been used for learning the prediction sets. For optimization, we use the combined loss as in (16) with

- log loss and $\widehat{\mathrm{CE}}_{KL}$ with $\gamma = 0.01$ for testing CE_{KL}
- Brier score and $\widehat{\mathrm{CE}}_2$ with $\gamma = 0.01$ for testing CE_2.

As optimisation parameters, we use a learning rate lr = 0.0001, a batch size = 256, the adam optimizer and train the neural network for 200 epochs.

Acknowledgements M.J. and W.W. received funding from the Flemish Government under the “Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen” programme.

Author contributions MJ conducted the experiments and wrote the main manuscript. MJ, TM and WW conceived the experiments and analyzed the results. All authors helped with conceptualization and reviewing the manuscript.

Data availability Data is provided within the manuscript or supplementary information files.

Declarations

Conflict of interest The authors declare no conflict of interest.

References

Abe, T., Buchanan, E. K., Pleiss, G., Zemel, R., & Cunningham, J. P. (2022). Deep ensembles work, but are they necessary? In *Proceedings of NeurIPS, 35th advances in neural information processing systems*.

Bengs, V., Hüllermeier, E., & Waegeman, W. (2022). Pitfalls of epistemic uncertainty quantification through loss minimisation. In *Proceedings of NeurIPS, 35th advances in neural information processing systems*.

Bengs, V., Hüllermeier, E., & Waegeman, W. (2023). On second-order scoring rules for epistemic uncertainty quantification. In *Proceedings of ICML, 40th international conference on machine learning*.

Brier, G. W. (1950). Verification of forecasts expressed in terms of probability. *Monthly Weather Review, 78*(1), 1.

Bröcker, J. (2009). Reliability, sufficiency, and the decomposition of proper scores. *Quarterly Journal of the Royal Meteorological Society, 135*, 1512–1519.

Bröcker, J., & Smith, L. A. (2007). Increasing the reliability of reliability diagrams. *Weather and Forecasting, 22*, 651–661.

Caprio, M., Dutta, S., Jang, K. J., Lin, V., Ivanov, R., Sokolsky, O., & Lee, I. (2024). Credal Bayesian deep learning. *Transactions on Machine Learning Research*.

Chau, S. L., Schrab, A., Gretton, A., Sejdinovic, D., & Muandet, K. (2024). Credal two-sample tests of epistemic ignorance. arXiv preprint arXiv:2410.12921.

Cozman, F. G. (2000). Credal networks. *Artificial Intelligence, 120*(2), 199–233.

Dümbgen, L. (2017). Empirische Prozesse. Institut für Mathematische Statistik und Versicherungsmathematik der Universität Bern, Bern, Switzerland. https://books.google.de/books?id=tmFMygEACAAJ.

Gal, Y., & Ghahramani, Z. (2016). Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of ICML, 32nd international conference on machine learning*.

Gelman, A., Carlin, J. B., Stern, H. S., & Rubin, D. B. (2004). *Bayesian data analysis* (2nd ed.). Boca Raton: Chapman and Hall.

Gneiting, T., & Raftery, A. E. (2007). Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association, 102*, 359–378.

Gruber, S., & Buettner, F. (2022). Better uncertainty calibration via proper scores for classification and beyond. In *Proceedings of NeurIPS, 35th advances in neural information processing systems*.

Gruber, C., Schenk, P. O., Schierholz, M., Kreuter, F., & Kauermann, G. (2023). Sources of uncertainty in machine learning—a statisticians' view. arXiv preprint arXiv:2305.16703.

Guo, C., Pleiss, G., Sun, Y., & Weinberger, K. Q. (2017). On calibration of modern neural networks. In *Proceedings of ICML, 34th international conference on machine learning*.

Gupta, K., Rahimi, A., Ajanthan, T., Mensink, T., Sminchisescu, C., & Hartley, R. (2020). Calibration of neural networks using splines. In *Proceedings of ICML, international conference on learning representations*.

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

Hosmer, D. W., Hosmer, T., Le Cessie, S., & Lemeshow, S. (1997). A comparison of goodness-of-fit tests for the logistic regression model. *Statistics in Medicine, 16*, 965–980.

Hüllermeier, E., Destercke, S., & Shaker, M. H. (2022). Quantification of credal uncertainty in machine learning: A critical analysis and empirical comparison. In *Proceedings of UAI, 34th conference on uncertainty in artificial intelligence*.

Hüllermeier, E., & Waegeman, W. (2021). Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine Learning, 110*, 457–506.

Javanmardi, A., Stutz, D., & Hüllermeier, E. (2024). Conformalized credal set predictors. In *Proceedings of NeurIPS, 37th advances in neural information processing systems*.

Juergens, M., Meinert, N., Bengs, V., Hüllermeier, E., & Waegeman, W. (2024). Is epistemic uncertainty faithfully represented by evidential deep learning methods? In *Proceedings of ICML, 41st international conference on machine learning*.

Kendall, A., & Gal, Y. (2017). What uncertainties do we need in bayesian deep learning for computer vision? In *Proceedings of NeurIPS, 30th advances in neural information processing systems*.

Kopetzki, A.-K., Charpentier, B., Zügner, D., Giri, S., & Günnemann, S. (2021). Evaluating robustness of predictive uncertainty estimation: Are Dirichlet-based models reliable? In *Proceedings of ICML, 38th international conference on machine learning*.

Krizhevsky, A., & Hinton, G. (2009). Learning multiple layers of features from tiny images.

Kull, M., & Flach, P. (2015). Novel decompositions of proper scoring rules for classification: Score adjustment as precursor to calibration. In *Machine learning and knowledge discovery in databases*.

Kumar, A., Sarawagi, S., & Jain, U. (2018). Trainable calibration measures for neural networks from kernel mean embeddings. In *Proceedings of ICML, 35th international conference on machine learning*.

Lakshminarayanan, B., Pritzel, A., & Blundell, C. (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. In *Proceedings of NeurIPS, 30th advances in neural information processing systems*.

Löhr, T., Hofman, P., Mohr, F., & Hüllermeier, E. (2025). Credal prediction based on relative likelihood. arXiv preprint arXiv:2505.22332.

Luo, R., Bhatnagar, A., Bai, Y., Zhao, S., Wang, H., Xiong, C., Savarese, S., Ermon, S., Schmerling, E., & Pavone, M. (2022). Local calibration: Metrics and recalibration. In *Proceedings of UAI, uncertainty in artificial intelligence*.

Marx, C., Zalouk, S., & Ermon, S. (2024). Calibration by distribution matching: Trainable kernel calibration metrics. In *Proceedings of NeurIPS, 36th advances in neural information processing systems*.

Meinert, N., Gawlikowski, J., & Lavin, A. (2023). The unreasonable effectiveness of deep evidential regression. In *Proceedings of AAAI, 37th proceedings of the AAAI conference on artificial intelligence*.

Mortier, T., Bengs, V., Hüllermeier, E., Luca, S., & Waegeman, W. (2023). On the calibration of probabilistic classifier sets. In *Proceedings of the 26th international conference on artificial intelligence and statistics*.

Müller, A. (1997). Integral probability metrics and their generating classes of functions. *Advances in Applied Probability, 29*, 429–443.

Murphy, A. H. (1973). A new vector partition of the probability score. *Journal of Applied Meteorology (1962–1982), 12*(4), 595–600.

Naeini, M. P., Cooper, G. F., & Hauskrecht, M. (2015). Obtaining well calibrated probabilities using Bayesian binning. In *Proceedings of AAAI, proceedings of the twenty-ninth AAAI conference on artificial intelligence*.

Nguyen, V.-L., Shaker, M. H., & Hüllermeier, E. (2022). How to measure uncertainty in uncertainty sampling for active learning. *Machine Learning, 111*, 89–122.

Niculescu-Mizil, A., & Caruana, R. (2005). Predicting good probabilities with supervised learning. In *Proceedings of ICML, 22nd international conference on machine learning*.

Ovadia, Y., Fertig, E., Ren, J., Nado, Z., Sculley, D., Nowozin, S., Dillon, J., Lakshminarayanan, B., & Snoek, J. (2019). Can you trust your model's uncertainty? Evaluating predictive uncertainty under dataset shift. In *Proceedings of NeurIPS, 31st advances in neural information processing systems*.

Popordanoska, T., Gregor Gruber, S., Tiulpin, A., Buettner, F., & Blaschko, M. B. (2024). Consistent and asymptotically unbiased estimation of proper calibration errors. In *Proceedings of the 27th international conference on artificial intelligence and statistics*.

Popordanoska, T., Sayer, R., & Blaschko, M. (2022). A consistent and differentiable lp canonical calibration error estimator.

Rahimi, A., Shaban, A., Cheng, C.-A., Hartley, R., & Boots, B. (2020). Intra order-preserving functions for calibration of multi-class neural networks. In *Proceedings of NeurIPS, 33rd advances in neural information processing systems*.

Sale, Y., Caprio, M., & Hüllermeier, E. (2023). Is the volume of a credal set a good measure for epistemic uncertainty? In *Proceedings of UAI, 39th conference on uncertainty in artificial intelligence*.

Senge, R., Bösner, S., Dembczyński, K., Haasenritter, J., Hirsch, O., Donner-Banzhoff, N., & Hüllermeier, E. (2014). Reliable classification: Learning classifiers that distinguish aleatoric and epistemic uncertainty. *Information Sciences, 255*, 16–29.

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition.

Ulmer, D., Hardmeier, C., & Frellsen, J. (2023). Prior and posterior networks: A survey on evidential deep learning methods for uncertainty estimation. *Transactions on Machine Learning Research*.

Vaart, A. W. (2000). *Asymptotic statistics*. Cambridge: Cambridge University Press.

Vaicenavicius, J., Widmann, D., Andersson, C., Lindsten, F., Roll, J., & Schön, T. (2019). Evaluating model calibration in classification. In *Proceedings of the twenty-second international conference on artificial intelligence and statistics*.

Walley, P. (1991). *Statistical reasoning with imprecise probabilities*. London: Chapman and Hall.

Wang, K., Shariatmadar, K., Manchingal, S. K., Cuzzolin, F., Moens, D., & Hallez, H. (2025). Creinns: credal-set interval neural networks for uncertainty estimation in classification tasks. In *Neural networks*.

Widmann, D., Lindsten, F., & Zachariah, D. (2019). Calibration tests in multi-class classification: A unifying framework. In *Proceedings of NeurIPS, 32nd advances in neural information processing systems*

Wolpert, D. H. (1992). Stacked generalization. *Neural Networks, 5*, 241–259.

Zadrozny, B., & Elkan, C. (2001). Obtaining calibrated probability estimates from decision trees and naive Bayesian classifiers. In *Proceedings of ICML, 18th international conference on machine learning*.

Zadrozny, B., & Elkan, C. (2002). Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of ACM SIGKDD, 8th international conference on knowledge discovery and data mining*.

Authors and Affiliations

Mira Jürgens[1] · Thomas Mortier[2] · Eyke Hüllermeier[3] · Viktor Bengs[3] · Willem Waegeman[1]

✉ Mira Jürgens
mira.juergens@ugent.be

Thomas Mortier
thomasf.mortier@ugent.be

Eyke Hüllermeier
eyke@ifi.lmu.de

Viktor Bengs
viktor.bengs@ifi.lmu.de

Willem Waegeman
willem.waegeman@ugent.be

[1] Department of Data Analysis and Mathematical Modeling, Ghent University, Coupure Links 653, 9000 Ghent, Belgium

[2] Department of Environment, Ghent University, Coupure Links 653, 9000 Ghent, Belgium

[3] Institute of Informatics, LMU Munich, Akademiestr. 7, 80799 Munich, Germany