MINIMAX BOOSTING WITH STRONG RULES

Anonymous authors
Paper under double-blind review

ABSTRACT

Current boosting methods excel at combining weak rules, such as decision trees, that are well-suited to certain types of tabular data. However, they cannot effectively combine strong rules, such as neural networks, which are required for data with a hierarchical structure. Existing boosting approaches generalize poorly with strong rules because they greedily minimize empirical losses, disregarding out-of-sample expected losses. This paper presents MBoost, a minimax approach for boosting that can effectively leverage strong rules. Unlike existing methods, MBoost minimizes worst-case expected losses and accounts for the generalization behavior of base rules. Both theoretical and experimental results show that MBoost methods can achieve excellent performance with strong base rules, overcoming the limitations of current approaches.

1 INTRODUCTION

Boosting is a principled approach to ensemble learning that can provide excellent practical performance with theoretical guarantees (see e.g., Schapire & Freund (2012)). In particular, current boosting methods are widely regarded as the preferred approach for tabular data that lack hierarchical structure (see e.g., Grinsztajn et al. (2022)). Most existing methods combine weak base rules such as decision trees. Indeed, boosting was originally conceived to answer the fundamental theoretical question of whether weak rules slightly better than random guesses can be combined to achieve high accuracy (Schapire, 1989; Freund & Schapire, 1996). A key factor in the success of existing boosting methods is that weak rules ensure small differences between expected and empirical losses (generalization gaps). However, weak rules are often inadequate for data with a hierarchical structure (e.g., digital images, biological signals, and natural language), which instead require strong rules such as neural networks (NNs) (see e.g., Mhaskar & Poggio (2016); Shen et al. (2019)).

Existing boosting methods cannot effectively combine strong base rules since their empirical losses are no longer dependable proxies of actual expected losses. These methods are prone to overfitting with strong rules because they select base rules based solely on empirical losses (margins of training samples) and disregard out-of-sample expected losses. The generalization gaps of strong rules can vary significantly, with some rules closely matching expected and empirical losses and others exhibiting substantial overfitting. The use of strong rules requires carefully accounting for generalization gaps. For example, standard training pipelines for NNs use data-driven estimates of expected losses (held-out validation estimates) to identify and filter out models with large generalization gaps.

This paper proposes MBoost, a minimax approach for boosting that effectively leverages strong rules. MBoost methods minimize worst-case expected losses instead of empirical losses, and account for the actual generalization behavior of individual base rules. Recent work has shown that minimax methods can improve the generalization of boosting techniques under corrupted samples (Mazuelas & Alvarez, 2025); here, we show that minimax formulations also enable the reliable combination of strong rules. Specifically, the main contributions presented in the paper are as follows.

- We develop tractable convex formulations for MBoost learning that reveal the key role played by the generalization gaps of base rules as L1 penalization terms.
- We present theoretical guarantees showing that MBoost can avoid overfitting by assessing generalization gaps and can provide performance approaching the Bayes risk.
- We present efficient algorithms for MBoost learning that use data-driven estimates of the generalization gaps for individual base rules.
- The experiments show that MBoost can reliably combine strong rules, outperforming current boosting approaches.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Notation: Calligraphic letters represent sets; bold lowercase letters represent vectors; for a vector $\mathbf{v}$, $\mathbf{v}^\top$ and v_k denote its transpose and kth component, and $|\mathbf{v}|$ denotes the vector given by component-wise absolute values; $\text{sign}(\cdot)$ denotes the sign of its argument; $\mathbb{I}\{\cdot\}$ denotes the indicator operator; $\|\cdot\|_1$ and $\|\cdot\|_\infty$ denote the 1-norm and the infinity norm of its argument, respectively; and $\mathbb{E}_{\mathrm{p}}\{\cdot\}$ denotes the expectation of its argument with respect to probability distribution p.

2 PRELIMINARIES

This section describes related methods, and details the addressed setting and terminology used.

2.1 RELATED WORK

Strong rules are well-suited for ensemble methods and their combination has been widely shown to improve classification accuracies and uncertainty assessments (see e.g., Lakshminarayanan et al. (2017); Nam et al. (2021)). Most methods for ensembles of strong rules promote diversity among base rules by varying their initializations, hyperparameters, or learning schedules (Lakshminarayanan et al., 2017; Huang et al., 2017), while others encourage diversity more directly by emphasizing examples where base rules disagree (Pagliardini et al., 2023; Ortega et al., 2022). These practical strategies produce heterogeneous ensembles but can result in combinations of somewhat redundant base rules since the ensemble is not learned by explicitly exploiting base rules' complementarities.

Boosting methods directly promote the diversity and complementarity of the combined base rules. Over multiple learning rounds, these methods update the learned ensemble by adding a new base rule with low empirical loss at the samples where previous base rules fail. However, conventional approaches for boosting are not designed to handle strong base rules and often lead to overfitting (Opitz & Maclin, 1999; Rambhatla et al., 2022). Most existing boosting approaches that use strong rules (Wang et al., 2021; Schwenk & Bengio, 2000; Moghimi et al., 2016) address overfitting through a simple shrinkage-based regularization that uniformly scales down the contribution of all base rules (Friedman, 2001). More sophisticated methods utilize a capacity-conscious L1-regularization, where each base rule is penalized according to the (Rademacher) complexity of the subfamily containing that rule (Cortes et al., 2014; 2019). These methods can provide performance guarantees and have been shown to be effective using base rules that are moderately more complex than those used by conventional boosting methods (deeper decision trees). However, they struggle to handle strong rules such as NNs, since capacity measures like Rademacher complexity are disproportionately large for common strong rules and fail to capture the actual generalization behavior of individual base rules. Existing boosting methods struggle to exploit the advantages of strong rules because they only aim to minimize empirical losses. Among the base rules with small empirical loss, current techniques fail to distinguish between rules with high expected loss and those with low expected loss.

2.2 PROBLEM FORMULATION

Classification rules assign labels in a finite set $\mathcal{Y}$ to instances in a Borel set $\mathcal{X} \subset \mathbb{R}^d$. As is commonly done in the boosting literature, in the following we consider binary classification problems, i.e., the set of labels is $\mathcal{Y} = \{-1, +1\}$. We denote by $\Delta(\mathcal{X} \times \mathcal{Y})$ the set of probability distributions on $\mathcal{X} \times \mathcal{Y}$, endowed with a suitable sigma-algebra, while the set of classification rules is denoted by $\mathrm{T}(\mathcal{X}, \mathcal{Y})$. For $\mathrm{p} \in \Delta(\mathcal{X} \times \mathcal{Y})$, we denote by $\mathrm{p}_x \in \Delta(\mathcal{X})$ the marginal distribution over $\mathcal{X}$ and by $\mathrm{p}(y|x)$ the conditional probability of label $y \in \mathcal{Y}$ given $x \in \mathcal{X}$. The underlying distribution of samples is denoted as $\mathrm{p}^* \in \Delta(\mathcal{X} \times \mathcal{Y})$ and the empirical distribution of n training samples is denoted as $\mathrm{p}_n \in \Delta(\mathcal{X} \times \mathcal{Y})$. For rule $\mathrm{h} \in \mathrm{T}(\mathcal{X}, \mathcal{Y})$, we denote by $\mathrm{h}(y|x)$ the probability assigned to label $y \in \mathcal{Y}$ for instance $x \in \mathcal{X}$ and by $\mathrm{h}(x) \in \mathcal{Y}$ the label predicted for instance $x \in \mathcal{X}$.

Supervised classification methods use n training samples $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ to learn a classification rule h with small risk $R(\mathrm{h})$, i.e., expected loss $R(\mathrm{h}) = \mathbb{E}_{\mathrm{p}^*}\{\ell(\mathrm{h}, (x, y))\}$. The smallest expected loss is referred to as the Bayes ℓ-risk, R_{Bayes}, that is

$$R_{\text{Bayes}} = \min_{\mathrm{h} \in \mathrm{T}(\mathcal{X}, \mathcal{Y})} \mathbb{E}_{\mathrm{p}^*}\{\ell(\mathrm{h}, (x, y))\}. \tag{1}$$

In the following, we consider the log-loss $\ell(\mathrm{h}, (x, y)) = -\log \mathrm{h}(y|x)$, while the boosting methods introduced can be extended to general losses, similarly to the developments in Mazuelas & Alvarez (2025); Mazuelas et al. (2022); Balsubramani & Freund (2016) for other learning scenarios.

We consider general sets of base rules $\mathcal{H} = \{\hbar_1, \hbar_2, \ldots, \hbar_T\} \subset \mathrm{T}(\mathcal{X}, \mathcal{Y})$. Conventional boosting methods consider families of weak rules, e.g., decision trees with a bounded number of nodes. On the other hand, the methods proposed consider base rules that are not constrained to be simple or contained in a particular family, e.g., NNs with assorted architectures. In what follows, we only assume that the set $\mathcal{H}$ of base rules is symmetric in the sense that $-\hbar \in \mathcal{H}$ whenever $\hbar \in \mathcal{H}$, which is satisfied for all the commonly used types of classification rules.

For each base rule $\hbar_k \in \mathcal{H}$, the error probability with respect to distribution p is denoted as

$$\mathrm{err}(\mathrm{p}, \hbar_k) = \mathbb{E}_{\mathrm{p}}\{\mathbb{I}\{\hbar_k(x) \neq y\}\} \tag{2}$$

so that the (actual) expected classification error of base rule $\hbar_k$ is given by $\mathrm{err}(\mathrm{p}^*, \hbar_k)$, and the empirical classification error is given by $\mathrm{err}(\mathrm{p}_n, \hbar_k)$. The generalization gap of base rule $\hbar_k \in \mathcal{H}$ denoted by λ_k^* corresponds to the difference between its expected and empirical errors

$$\lambda_k^* = \Big|\mathrm{err}(\mathrm{p}^*, \hbar_k) - \mathrm{err}(\mathrm{p}_n, \hbar_k)\Big| \tag{3}$$

that is given by $\lambda_k^* = \mathrm{err}(\mathrm{p}^*, \hbar_k) - \mathrm{err}(\mathrm{p}_n, \hbar_k)$ in the usual case where the rule is obtained by fitting training samples. In addition, we denote by λ_k an estimate of the actual generalization gap λ_k^*.

The generalization gaps of weak rules included in families with reduced complexity like decision trees are small for all the base rules in the family. However, for more general types of base rules like NNs, the generalization gaps can vary significantly for different base rules, and take large values for base rules that significantly overfit. Generalization gaps are rather small for most base rules; for instance, the generalization gap is lower than $1.36/\sqrt{n}$ with probability at least 0.95 for a base rule randomly drawn from (a possibly highly complex) family $\mathcal{H}$. On the other hand, the generalization gap is large for certain base rules if family $\mathcal{H}$ has large Rademacher complexity.

The generalization gaps also change depending on the underlying distribution. If samples follow a distribution very close to the empirical distribution, generalization gaps remain small, as expected errors are always similar to empirical errors. On the other hand, for distributions with most of their mass far from the training instances, the generalization gaps are large. Therefore, the generalization gaps also provide information about the underlying distribution. Specifically, the set of probability distributions p consistent with a generalization gap up to λ_k for base rule $\hbar_k$ is given by

$$\Big|\mathrm{err}(\mathrm{p}, \hbar_k) - \mathrm{err}(\mathrm{p}_n, \hbar_k)\Big| \leq \lambda_k \Leftrightarrow \mathrm{err}(\mathrm{p}_n, \hbar_k) - \lambda_k \leq \mathrm{err}(\mathrm{p}, \hbar_k) \leq \mathrm{err}(\mathrm{p}_n, \hbar_k) + \lambda_k \tag{4}$$

that corresponds to a linear stripe of width $2\lambda_k$ in the space of probability distributions $\Delta(\mathcal{X} \times \mathcal{Y})$.

The usage of weak rules enables existing boosting methods to select base rules based solely on empirical losses. The following presents boosting methods that effectively leverage strong rules by accounting for their individual generalization gaps.

3 Minimax boosting with general base rules

MBoost methods learn classification rules by solving the minimax problem

$$\min_{\mathrm{h} \in \mathrm{T}(\mathcal{X}, \mathcal{Y})} \max_{\mathrm{p} \in \mathcal{V}} \mathbb{E}_{\mathrm{p}}\{\ell(\mathrm{h}, (x, y))\} \tag{5}$$

where the uncertainty set of probability distributions $\mathcal{V}$ is given by

$$\mathcal{V} = \Big\{\mathrm{p} \in \Delta(\mathcal{X} \times \mathcal{Y}) : \ \mathrm{p}_x = \mathrm{p}_x^* \ \text{ and } \ \Big|\mathrm{err}(\mathrm{p}, \hbar_k) - \mathrm{err}(\mathrm{p}_n, \hbar_k)\Big| \leq \lambda_k \ \forall\, \hbar_k \in \mathcal{H}\Big\}. \tag{6}$$

Conventional boosting approaches minimize empirical losses, relying solely on the empirical distribution of training samples. In contrast, the proposed approach (optimization in (5)) considers multiple distributions in uncertainty set $\mathcal{V}$, leading to improved robustness against overfitting. The formulation in (5) belongs to the class of minimax approaches commonly referred to as robust risk minimization or distributionally robust techniques (Lee & Raginsky, 2018; Mazuelas et al., 2023; 2020; Shafieezadeh-Abadeh et al., 2019; Duchi & Namkoong, 2019), which rely on uncertainty sets composed of plausible distributions. Common methods regard probability distributions as plausible if they are close to the empirical distribution of training samples in terms of metrics such as Wasserstein distance (Lee & Raginsky, 2018) or f-divergence (Duchi & Namkoong, 2019).

The methods proposed consider uncertainty sets of distributions tailored to combining general base rules while avoiding overfitting. In particular, MBoost methods regard distributions as plausible if

they are consistent with empirical errors and estimated generalization gaps. Therefore, the proposed methods are not affected by overfitting as long as the generalization gaps are reliably estimated. For instance, if the estimated generalization gaps are accurate ($\lambda_k = \lambda_k^*$ for any $\hbar_k \in \mathcal{H}$), then $\mathrm{p}^* \in \mathcal{V}$ and the risk of the learned MBoost is upper bounded by the minimax value in (5). Furthermore, as shown in Sections 4 and 6, the methods presented remain resilient to overfitting using approximate estimates for the generalization gaps.

For notational convenience, we use vector notation in the following and denote by $\boldsymbol{\hbar}(x)$ and $\boldsymbol{\lambda}$ the vectors in $\mathbb{R}^T$ with kth component given by the kth base rule prediction $\hbar_k(x)$ and estimated generalization gap λ_k. In addition, we denote by $R_{\boldsymbol{\lambda}}$ the minimax value of (5) and by $\mathbf{e} \in \mathbb{R}^T$ the vector given by the empirical classification errors of base rules, that is

$$\mathbf{e} = [\mathrm{err}(\mathrm{p}_n, \hbar_1), \mathrm{err}(\mathrm{p}_n, \hbar_2), \ldots, \mathrm{err}(\mathrm{p}_n, \hbar_T)]^\top, \text{ with } \mathrm{err}(\mathrm{p}_n, \hbar_k) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}\{\hbar_k(x_i) \neq y_i\}. \quad (7)$$

At first sight, the minimax problem in (5) addressed by MBoost may seem to be computationally prohibitive in practice. The next result shows that MBoost classification rules can be learned by solving the tractable convex optimization problem

$$\min_{\boldsymbol{\mu}} \ \mathbf{e}^\top \boldsymbol{\mu} + \mathbb{E}_{\mathrm{p}_x^*}\{\log \psi(x, \boldsymbol{\mu})\} + \boldsymbol{\lambda}^\top |\boldsymbol{\mu}| \quad (8)$$

with

$$\psi(x, \boldsymbol{\mu}) = \exp\Big(-\sum_{k:\hbar_k(x)=1} \mu_k\Big) + \exp\Big(-\sum_{k:\hbar_k(x)=-1} \mu_k\Big). \quad (9)$$

Theorem 1. Let $\mathcal{V}$ be an uncertainty set given by equation 6 that has non-empty interior, i.e., there exists some $\mathrm{p} \in \mathcal{V}$ such that $|\mathrm{err}(\mathrm{p}, \hbar_k) - \mathrm{err}(\mathrm{p}_n, \hbar_k)| < \lambda_k$ for any $\hbar_k \in \mathcal{H}$. If $\boldsymbol{\mu}_{\boldsymbol{\lambda}}$ is a solution of optimization problem (8), the classification rule

$$\mathrm{h}^{\boldsymbol{\mu}_{\boldsymbol{\lambda}}}(y|x) = \big(1 + \exp(-y\,\boldsymbol{\hbar}(x)^\top \boldsymbol{\mu}_{\boldsymbol{\lambda}})\big)^{-1} \quad (10)$$

is a solution of the minimax problem (5) and the minimax value is given by

$$R_{\boldsymbol{\lambda}} = \mathbf{e}^\top \boldsymbol{\mu}_{\boldsymbol{\lambda}} + \mathbb{E}_{\mathrm{p}_x^*}\{\log \psi(x, \boldsymbol{\mu}_{\boldsymbol{\lambda}})\} + \boldsymbol{\lambda}^\top |\boldsymbol{\mu}_{\boldsymbol{\lambda}}|. \quad (11)$$

Furthermore, the minimax rule $\mathrm{h}^{\boldsymbol{\mu}_{\boldsymbol{\lambda}}}$ corresponds to the worst-case distribution in $\mathcal{V}$, that is, $\mathrm{p}^{\boldsymbol{\mu}_{\boldsymbol{\lambda}}}(x, y) \coloneqq \mathrm{p}_x^*(x)\mathrm{h}^{\boldsymbol{\mu}_{\boldsymbol{\lambda}}}(y|x)$ is the distribution in $\mathcal{V}$ that has the largest Bayes ℓ-risk, i.e.,

$$\mathrm{p}^{\boldsymbol{\mu}_{\boldsymbol{\lambda}}}(x, y) \in \arg\max_{\mathrm{p} \in \mathcal{V}} \min_{\mathrm{h} \in \mathrm{T}(\mathcal{X}, \mathcal{Y})} \mathbb{E}_{\mathrm{p}}\{\ell(\mathrm{h}, (x, y))\}. \quad (12)$$

Proof. See Appendix B. □

The result above shows how to learn MBoost rules that solve the minimax problem in (5). Firstly, the theorem characterizes minimax rules as linear combinations of base rules given by

$$\mathrm{h}^{\boldsymbol{\mu}}(y|x) = \big(1 + \exp(-y\,\boldsymbol{\hbar}(x)^\top \boldsymbol{\mu})\big)^{-1}, \ \mathrm{h}^{\boldsymbol{\mu}}(x) = \mathrm{sign}(\boldsymbol{\hbar}(x)^\top \boldsymbol{\mu}) \quad (13)$$

so that predictions correspond to weighted votes from base rules, and probability assessments are given by margins $(y\,\boldsymbol{\hbar}(x)^\top \boldsymbol{\mu})$ using the sigmoid link function. Secondly, Theorem 1 shows that the parameters of MBoost rules can be learned by solving the composite convex optimization in (8), and that MBoost rules correspond to a worst-case assessment of the uncertainty present.

The minimization in (8) has an insightful interpretation. The first term, $\mathbf{e}^\top \boldsymbol{\mu}$, encodes the supervised information and encourages base rules with low empirical error.[1] The second term, $\mathbb{E}_{\mathrm{p}_x^*}\{\log \psi(x, \boldsymbol{\mu})\}$, acts as an unsupervised loss that penalizes overconfident ensemble predictions and promotes diversity among base rules. Specifically, this term penalizes ensembles whose votes concentrate on the class with a majority of votes, since

$$\mathbb{E}_{\mathrm{p}_x^*}\{\log \psi(x, \boldsymbol{\mu})\} \approx \mathbb{E}_{\mathrm{p}_x^*}\Big\{\max_{y \in \mathcal{Y}} \sum_{k:\hbar_k(x)=y} -\mu_k\Big\} = -\mathbb{E}_{\mathrm{p}_x^*}\Big\{\min_{y \in \mathcal{Y}} \sum_{k:\hbar_k(x)=y} \mu_k\Big\} \quad (14)$$

so that $\mathbb{E}_{\mathrm{p}_x^*}\{\log \psi(x, \boldsymbol{\mu})\}$ acts as a smooth approximation of (minus) the expected score (vote) for the minority class.

[1] Notice that MBoost parameters $\boldsymbol{\mu}$ can be taken to be nonnegative since $\mathcal{H}$ is symmetric.

The minimization in (8) also reveals the key role played by estimates of generalization gaps. Specifically, the third term, $\boldsymbol{\lambda}^\top|\boldsymbol{\mu}|$, corresponds to an L1-regularization that discourages base rules with a high generalization gap. Together, the first and third terms encourage base rules with low out-of-sample expected error since they account for the sum of empirical error and generalization gap. The effect of these two terms resembles standard machine learning pipelines: 1) learn base rules with low empirical error; 2) estimate the generalization gaps of the base rules learned (e.g., through held-out validation estimates); and 3) select a rule with an adequate balance between empirical error and generalization gap (expected error). Overall, the objective in (8) combines base rules by jointly accounting for expected error, prediction diversity, and estimated uncertainty.

The optimization problem in (8) is composite convex (sum of a smooth convex function with a decomposable L1 penalization) and amenable to conventional efficient optimization methods such as stochastic gradient descent. (Sub)gradients for the first and third terms in (8) can be obtained straightforwardly and those for the second term can be accurately approximated even with unlabeled samples since such a term is unsupervised. Indeed, the gradient of the second term has an insightful interpretation that complements the discussion above. Specifically, we have

$$\begin{aligned} -\nabla_k(\mathbb{E}_{\mathrm{p}^*_x}\{\log\psi(x,\boldsymbol{\mu})\}) &= \mathbb{E}_{\mathrm{p}^*_x}\left\{\left(1+\exp(\hbar_k(x)\boldsymbol{\hbar}(x)^\top\boldsymbol{\mu})\right)^{-1}\right\} \\ &= \mathbb{E}_{\mathrm{p}^*_x}\left\{\mathrm{h}^{\boldsymbol{\mu}}(y\neq\hbar_k(x)|x)\right\} = \mathrm{err}(\mathrm{p}^{\boldsymbol{\mu}},\hbar_k). \end{aligned} \tag{15}$$

Therefore, in gradient descent iterations, the second term in (8) increases the coefficients of base rules with a significant average disagreement with the current minimax rule $\mathrm{h}^{\boldsymbol{\mu}}$, which corresponds to a significant expected error with respect to the current worst-case distribution $\mathrm{p}^{\boldsymbol{\mu}}$. Hence, such a term results in gradient steps that promote base rules that differ from the previous ensemble in the instances previously predicted with high confidence.

MBoost methods can be directly compared to existing boosting approaches that use regularization to penalize complex ensembles and base rules (deeper decision trees). Current methods either apply a uniform penalization to all the base rules (i.e., $\lambda_k = \Lambda \geq 0$ for any k (Schapire & Freund, 2012; Duchi & Singer, 2009)) or assign the same penalization within predefined subfamilies (e.g., trees with the same depth). In the latter case, capacity-aware penalties of the form $\lambda_k = \Lambda_0 + \Lambda_1\mathcal{R}(\mathcal{H}_k)$ are used (Cortes et al., 2014; 2019), where $\mathcal{R}(\mathcal{H}_k)$ measures the capacity (Rademacher complexity) of the subfamily containing the kth base rule. While effective for relatively simple base rules, these approaches are not suitable for general strong rules. First, standard capacity measures can be excessively large and fail to capture the generalization behavior of common families (e.g., the Rademacher complexity of NNs increases exponentially with the network depth). Second, uniform or subfamily-level penalization is too coarse for strong rules, as generalization gaps vary significantly across individual rules. Finally, existing methods set specific penalization intensities (values of $\Lambda, \Lambda_0, \Lambda_1$) through cross-validation strategies, which are often computationally prohibitive with strong rules. The proposed approach addresses these limitations by using data-driven estimates of the penalization strengths $\boldsymbol{\lambda}$ (generalization gaps) of individual base rules, as detailed in Section 5. MBoost methods can effectively leverage strong rules by heavily penalizing those that actually overfit in the specific data distribution present, while allowing the unrestricted usage of the others.

As with other boosting methods (see e.g., Schapire & Freund (2012); Cortes et al. (2019)), the minimization in (8) is especially well-suited for coordinate descent approaches, as detailed in Section 5. Coordinate-wise, the first two terms in (8) are Lipschitz-continuous and the third term is a simple absolute-value penalty. Furthermore, the results below show that the methods presented are not heavily affected by the usage of efficient approximate optimization methods.

4 Theoretical analysis

The next theorem provides generalization bounds for MBoost methods in terms of the estimates used for generalization gaps and the quantity optimized during learning, i.e., the minimax risk $R_{\boldsymbol{\lambda}}$.

Theorem 2. Let $\mathcal{V}$ be an uncertainty set given by equation 6 that has non-empty interior, $\boldsymbol{\mu}_{\boldsymbol{\lambda}}$ be a solution of optimization problem (8), and $\boldsymbol{\lambda}^*$ be the actual generalization gaps of base rules as given in equation 3. The following statements hold.

1) The risk of the minimax rule $\mathrm{h}^{\boldsymbol{\mu}_{\boldsymbol{\lambda}}}$ in equation 10 is upper-bounded by the minimax risk $R_{\boldsymbol{\lambda}}$ as

$$R(\mathrm{h}^{\boldsymbol{\mu}_{\boldsymbol{\lambda}}}) \leq R_{\boldsymbol{\lambda}} + (\boldsymbol{\lambda}^* - \boldsymbol{\lambda})^\top|\boldsymbol{\mu}_{\boldsymbol{\lambda}}| \leq R_{\boldsymbol{\lambda}} + \|\boldsymbol{\lambda}^* - \boldsymbol{\lambda}\|_\infty\|\boldsymbol{\mu}_{\boldsymbol{\lambda}}\|_1. \tag{16}$$

2) The best possible upper bound in (16) is achieved using perfect estimates for the generalization gaps ($\boldsymbol{\lambda} = \boldsymbol{\lambda}^*$). Specifically, we have

$$R_{\boldsymbol{\lambda}^*} \leq R_{\boldsymbol{\lambda}} + (\boldsymbol{\lambda}^* - \boldsymbol{\lambda})^\top |\boldsymbol{\mu}_{\boldsymbol{\lambda}}| \tag{17}$$

and if the uncertainty set given by equation 6 has non-empty interior when $\boldsymbol{\lambda} = \boldsymbol{\lambda}^*$, we have

$$\left(R_{\boldsymbol{\lambda}} + (\boldsymbol{\lambda}^* - \boldsymbol{\lambda})^\top |\boldsymbol{\mu}_{\boldsymbol{\lambda}}|\right) - R_{\boldsymbol{\lambda}^*} \leq |\boldsymbol{\lambda}^* - \boldsymbol{\lambda}|^\top |\boldsymbol{\mu}_{\boldsymbol{\lambda}^*} - \boldsymbol{\mu}_{\boldsymbol{\lambda}}| \leq \|\boldsymbol{\lambda}^* - \boldsymbol{\lambda}\|_\infty \|\boldsymbol{\mu}_{\boldsymbol{\lambda}^*} - \boldsymbol{\mu}_{\boldsymbol{\lambda}}\|_1. \tag{18}$$

3) If $\boldsymbol{\mu}$ is an ε_{opt}-optimal solution of (8), that is

$$\mathbf{e}^\top \boldsymbol{\mu} + \mathbb{E}_{\mathrm{p}_x^*}\{\log \psi(x, \boldsymbol{\mu})\} + \boldsymbol{\lambda}^\top |\boldsymbol{\mu}| \leq R_{\boldsymbol{\lambda}} + \varepsilon_{\text{opt}} \tag{19}$$

we have

$$R(\mathrm{h}^{\boldsymbol{\mu}}) \leq R_{\boldsymbol{\lambda}} + (\boldsymbol{\lambda}^* - \boldsymbol{\lambda})^\top |\boldsymbol{\mu}| + \varepsilon_{\text{opt}}. \tag{20}$$

Proof. See Appendix C. □

The result above provides performance bounds for MBoost methods that characterize the impact of approximate generalization gaps and optimization algorithms.

The first part of Theorem 2 shows generalization bounds for MBoost methods. In particular, the minimax risk $R_{\boldsymbol{\lambda}}$ minimized during learning is an upper bound for MBoost risk if the estimates for generalization gaps $\boldsymbol{\lambda}$ upper bound the actual generalization gaps $\boldsymbol{\lambda}^*$. Furthermore, the bound in equation 16 shows that MBoost risk is controlled by the minimax risk as long as the estimates $\boldsymbol{\lambda}$ are not significantly lower than the actual generalization gaps $\boldsymbol{\lambda}^*$. For instance, for an MBoost that combines K base rules ($\mu_k \neq 0$ only for K components) and estimates generalization gaps using n_{val} fresh validation samples, the excess risk is of the order $\mathcal{O}(\|\boldsymbol{\mu}\|_1 \sqrt{\log(K)/n_{\text{val}}})$ as a direct consequence of Hoeffding's inequality.

The second part of the theorem further characterizes the dependence of the generalization bounds on the accuracy of the estimates for generalization gaps. In particular, the bound in equation 17 shows that the lowest generalization bound is achieved by using perfect estimates for generalization gaps. In addition, the bound in equation 18 shows that MBoost methods can effectively utilize approximate estimates for generalization gaps since the performance guarantees degrade gracefully as $\boldsymbol{\lambda}$ departs from $\boldsymbol{\lambda}^*$.

The figure on the right illustrates the impact of estimates for generalization gaps on the performance of MBoost in practice (see Appendix F.1 for a detailed description of the experimentation carried out). In accordance with the discussion above, MBoost provides strong performance as long as the estimated generalization gaps are not far from actual values. In particular, the figure shows that the data-driven estimates presented in Section 5 result in performance close to that achieved with access to the actual generalization gaps. In contrast, conventional regularization strategies with constant penalizations for base rules result in poor performance when using strong rules, even if penalization parameters are carefully tuned via cross-validation.

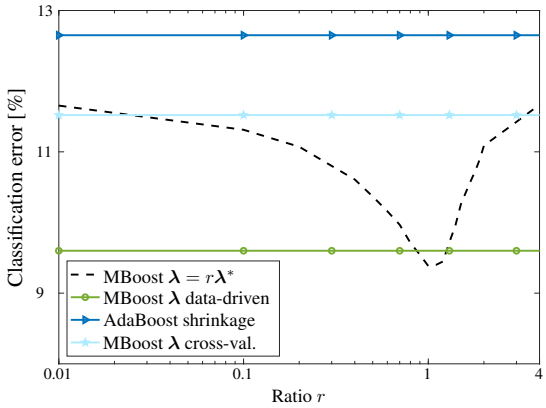


Figure 1: Comparison of classification errors for MBoost with different choices of estimated generalization gaps $\boldsymbol{\lambda}$ on the CIFAR-10 'cat vs dog' dataset. The figure also shows the performance that would be achieved using estimates proportional to the actual generalization gaps (black dashed curve).

The third part of the theorem shows that optimization (8) does not need to be solved with high accuracy since optimization errors only result in an additive term for the generalization bound. Therefore, efficient approximate optimization algorithms can be safely used in practice for MBoost learning.

The performance guarantees described in Theorem 2 can be compared with those for existing boosting methods. In particular, upper-bounding the generalization gap λ_k^* of each base rule with the Rademacher complexity of the set of base rules $\mathcal{R}(\mathcal{H})$, the generalization bounds in Theorem 2

show an excess risk of order $\mathcal{O}(\mathcal{R}(\mathcal{H})\|\boldsymbol{\mu}\|_1)$ in agreement with existing bounds for L1-regularized boosting (Cortes et al., 2014; 2019). As described above, bounds based on Rademacher complexity are appropriate for families of rules with reduced capacity but become loose for strong rules as NNs.

The above theoretical guarantees detail MBoost generalization capabilities using general estimates for generalization gaps. The next theorem provides performance bounds in terms of the Bayes risk.

Theorem 3. Let $\boldsymbol{\mu}_{\mathrm{B}}$ be coefficients for base rules such that

$$\mathbb{E}_{\mathrm{p}^*_x}\left\{\mathrm{kl}\big(\mathrm{p}^*(1|x)||\sigma(\boldsymbol{\hbar}(x)^\top\boldsymbol{\mu}_{\mathrm{B}})\big)\right\} \leq \varepsilon_{\text{approx}} \tag{21}$$

for $\mathrm{kl}(\cdot||\cdot)$ denoting the binary Kullback-Leibler divergence and $\sigma(\cdot)$ denoting the sigmoid function. If $\boldsymbol{\mu}$ is an ε_{opt}-optimal solution of (8), we have

$$R(\mathrm{h}^{\boldsymbol{\mu}}) \leq R_{\text{Bayes}} + (\boldsymbol{\lambda}^* + \boldsymbol{\lambda})^\top|\boldsymbol{\mu}_{\mathrm{B}}| + (\boldsymbol{\lambda}^* - \boldsymbol{\lambda})^\top|\boldsymbol{\mu}| + \varepsilon_{\text{approx}} + \varepsilon_{\text{opt}}. \tag{22}$$

Proof. See Appendix D. □

The result above shows that MBoost can achieve performance approaching the Bayes risk using base rules that can approximate optimal classification rules. This result can be compared with existing theoretical analyses of the Bayes consistency of boosting methods (Lugosi & Vayatis, 2004; Bartlett & Traskin, 2007). In particular, cases where $\varepsilon_{\text{approx}}$ in equation 21 can be made negligible correspond to the denseness assumption in Lugosi & Vayatis (2004), satisfied, for instance, by decision trees with $d+1$ nodes in $\mathbb{R}^d$. The regularization in Lugosi & Vayatis (2004) corresponds to the penalization $\lambda_k = \Lambda$ for all base rules, with $\Lambda \to 0$ and $\Lambda\sqrt{n/\log(n)} \to \infty$ as n tends to infinity. Such conditions result in $(\boldsymbol{\lambda}^* + \boldsymbol{\lambda})^\top|\boldsymbol{\mu}_{\mathrm{B}}| + (\boldsymbol{\lambda}^* - \boldsymbol{\lambda})^\top|\boldsymbol{\mu}| \leq o(1)$ as $n \to \infty$ for any family $\mathcal{H}$ with finite VC dimension, since in that case the generalization gaps are of order $\mathcal{O}(\sqrt{\log(n)/n})$.

Previous analyses of Bayes consistency consider weak rules such as decision trees, whereas the results and methods presented can leverage general base rules such as NNs, which often approximate functions more efficiently (Mhaskar & Poggio, 2016; Shen et al., 2019; DeVore et al., 2021). In particular, strong rules can enable accurate approximations using more concise combinations of base rules (i.e., smaller $\|\boldsymbol{\mu}\|_1$), especially in domains with hierarchically structured data where tree-based rules are less suitable.

The results presented in this section show that MBoost can leverage the strengths of strong rules and effectively avoid overfitting. The next section presents efficient algorithms for MBoost learning.

5 Learning algorithm

The proposed algorithm for MBoost learning solves the optimization problem in (8) using a coordinate descent approach, as detailed in Algorithm 1 and described in the following.

Approximate Gauss-Southwell (GS) coordinate descent is carried out using a base-learner that fits weighted samples (e.g., an off-the-shelf learner of NNs). Appendix E shows that the weights in line 10 of Alg. 1 correspond to descent coordinates that (approximately) maximize the magnitude of the gradient for the objective in (8). Coordinate descent with approximate GS updates is the most common optimization approach for boosting. In particular, gradient boosting in the functional space of combinations of base rules can also be seen as coordinate descent in the space of combination coefficients (Cortes et al., 2019). In addition, these methods can leverage efficient algorithms that roughly maximize the descent gradients since the iteration complexity of approximate GS worsens gracefully with the suboptimality of the selected coordinate (Nutini et al., 2015; Karimireddy et al., 2019).

Algorithm 1 MBoost learning

Input: Training samples $\{(x_i, y_i)\}_{i=1}^n$
parameters n_{held} (held-out size) and K (rounds)
Output: Coefficients $\boldsymbol{\mu} \in \mathbb{R}^K$, base rules $\hbar_1, \hbar_2, \ldots, \hbar_K$

1: $\{(x'_i, y'_i)\}_{i=1}^n \leftarrow \text{RandomShuffle}(\{(x_i, y_i)\}_{i=1}^n)$
2: $\mathbf{w} \leftarrow \mathbf{1} \in \mathbb{R}^{n-n_{\text{held}}}$
3: **for** $k = 1, 2, \ldots, K$ **do**
4: $\quad \hbar_k \leftarrow \text{BaseLearner}\big(\mathcal{H}, \{(x'_i, y'_i, w_i)\}_{i=1}^{n-n_{\text{held}}}\big)$
5: $\quad e_k \leftarrow \frac{1}{n}\sum_{i=1}^{n}\mathbb{I}\{\hbar_k(x'_i) \neq y'_i\}$
6: $\quad \lambda_k \leftarrow \left|e_k - \frac{1}{n_{\text{held}}}\sum_{i=n-n_{\text{held}}+1}^{n}\mathbb{I}\{\hbar_k(x'_i) \neq y'_i\}\right|$
7: $\quad \boldsymbol{\mu} \leftarrow$ solve (8) for $\boldsymbol{\hbar}(x) = [\hbar_1(x), \hbar_2(x), \ldots, \hbar_k(x)]^\top$
$\qquad$ using the training instances $\{x_i\}$
8: $\quad \{(x'_i, y'_i)\}_{i=1}^n \leftarrow \text{RandomShuffle}(\{(x_i, y_i)\}_{i=1}^n)$
9: $\quad$ **for** $i = 1, 2, \ldots, n - n_{\text{held}}$ **do**
10: $\quad\quad w_i \leftarrow 1 - \mathrm{h}^{\boldsymbol{\mu}}(y'_i|x'_i) = \big(1 + \exp(y'_i\boldsymbol{\hbar}(x'_i)^\top\boldsymbol{\mu})\big)^{-1}$

Algorithm 1 uses data-driven estimates for the generalization gaps of individual base rules. In particular, the generalization gap of the base rule selected at each round is assessed using a simple approach based on held-out estimates. Specifically, a subset of samples randomly drawn at each round is used by the base-learner (line 4 in Alg. 1), while the remaining samples are used to estimate the generalization gap (line 6 in Alg. 1). Intuitively, these held-out estimates of generalization gaps are accurate because different samples are used to learn the base rule and to estimate its classification error. The experimental results in Appendix F.4 further validate the reliability of these data-driven estimates of generalization gaps. Their estimation error is difficult to analyze theoretically, since samples are reused across boosting rounds. The use of base-learners with random subsampling is also common in existing boosting methods (Chen & Guestrin, 2016; Ke et al., 2017). In particular, subsampling strategies have been shown to improve efficiency and robustness without performance degradation (Friedman, 2002). A novel aspect of the methods presented is the use of held-out estimates for base-rule errors. While held-out estimates are common in other ensemble methods, such as random forests (Hastie et al., 2009), their use in boosting has been largely underexplored and limited to early stopping (Scikit-learn library, 2025).

The computational complexity of Algorithm 1 is very similar to that of conventional boosting methods. The main differences with standard methods lie in line 6 for generalization gap estimates (with negligible computational cost), and line 7, which carries out totally corrective updates. The use of totally corrective updates is common in certain boosting methods (Warmuth et al., 2006; Shen et al., 2013) and can result in steeper decreases in the optimization objective that lead to fewer boosting rounds and hence increased efficiency. This is especially important when using strong rules since the calls to the base-learner (line 4 in Alg. 1) are significantly more computationally demanding than the totally corrective step (line 7 in Alg. 1). In particular, for $k \leq K$ selected base rules, the totally corrective step obtains the MBoost parameters by solving the convex optimization problem in (8) with k variables. This optimization can be carried out very efficiently using conventional techniques for smooth convex optimization (logistic) with L1-regularization (Defazio et al., 2014).

6 Experimental results

The experiments compare the classification performance obtained by MBoost with that of state-of-the-art approaches, including standard SGD-based learning, (deep) ensemble techniques, and existing boosting methods for NNs. Due to the extensive methodological results presented in the preceding sections, this section remains necessarily concise. Appendix F provides all the implementation details as well as multiple additional experimental results, including calibration plots and detailed running times. In addition, the implementation code is available in the supplementary material.

Table 1: Average classification error (left sub-table) and ECE (right sub-table) in % ± standard deviation for MBoost and state-of-the-art methods using multiple real-world datasets. The upper sub-table shows results on datasets formed by images and the lower sub-table shows results on tabular data.

Dataset	Baseline	AdaBoost	Ensemble	MBoost	Baseline	AdaBoost	Ensemble	MBoost
Diabetic R.	3.73±.82	3.99±1.4	6.73±.50	**3.29±.98**	3.33±.63	3.80±1.4	4.10±.50	**1.64±.56**
R vs F Face	37.7±1.5	40.8±3.4	37.2±1.7	**36.6±2.5**	9.53±2.0	27.4±.69	**6.80±.25**	9.62±2.7
Cat vs dog	9.31±.52	12.5±1.5	9.29±.42	**9.17±.82**	**2.09±.49**	10.4±1.4	2.33±.70	2.69±.50
Car vs truck	3.50±.54	4.75±.73	4.26±.24	**3.33±.49**	1.20±.20	4.63±.68	1.53±.76	**1.02±.39**
Food101	3.95±1.5	3.30±1.4	3.05±.16	**2.60±.88**	2.71±1.0	3.27±1.3	4.19±.97	**1.61±.46**
MNIST	1.60±.44	**1.35±.15**	1.68±.20	1.56±.41	1.38±.21	1.25±.12	2.66±.06	**.54±.22**
F. MNIST	8.13±.93	9.64±.51	9.95±.24	**7.79±.65**	1.99±.69	7.42±.67	3.17±.19	**1.70±.29**
Breast H.	.84±.41	.61±.22	.87±.22	**.55±.32**	.80±.13	.60±.22	1.29±.10	**.44±.32**
Chest	2.19±.63	1.82±.50	4.37±.48	**1.79±.77**	1.81±.63	1.76±.48	2.41±.47	**1.05±.33**
Avg Time	1	1.07	2.74	1.08	1	1.07	2.74	1.08
Ion	11.4±5.7	10.6±3.0	8.89±5.4	**8.57±4.5**	10.9±5.3	10.6±2.9	9.60±3.1	**7.83±2.8**
Heart	16.2±4.8	19.6±9.3	19.3±7.4	**13.7±6.3**	15.5±2.8	19.0±8.8	**15.4±4.7**	**15.4±3.6**
Diabetes	22.8±4.0	26.7±3.4	27.9±5.0	**21.8±5.4**	13.0±2.4	24.8±3.4	15.3±3.5	**12.6±3.2**
German N.	24.2±4.2	27.2±5.6	**23.2±1.6**	23.9±4.6	9.58±2.4	25.1±4.5	12.2±3.4	**9.37±2.3**
Credit	14.1±1.8	17.1±3.6	14.6±2.8	**12.5±2.9**	10.4±1.6	17.3±4.7	10.3±2.7	**9.91±2.0**
Raisin	13.4±3.0	15.4±3.2	13.7±3.6	**12.6±2.8**	10.4±1.4	15.4±3.2	9.93±.89	**9.48±1.9**
Blood T.	22.7±3.2	26.8±4.6	28.2±4.4	**22.4±3.9**	18.4±2.3	21.2±5.1	17.5±4.4	**12.2±2.0**
QSAR	14.1±2.9	13.2±2.5	10.9±3.4	**10.3±2.3**	9.22±4.4	12.7±2.4	7.73±2.2	**7.28±1.2**
Haberman	29.3±1.9	31.0±5.5	32.3±5.7	**26.7±5.2**	15.5±8.8	30.2±6.8	21.5±4.2	**15.1±5.6**
Avg Time	1	17.6	16.9	15.3	1	17.6	16.9	15.3

Table 1 reports the performance achieved by MBoost in comparison with benchmark methods across 18 datasets: nine common image datasets and nine tabular datasets known to be well-suited for NNs. For the image datasets, all the methods use a ResNet-50 backbone trained with SGD (common learning rates and momentum) for 100 epochs. In particular, ensembles combine $K = 10$ base rules learned over 10 epochs. For the tabular datasets, all the methods use automated model selection of feedforward fully connected NNs based on Bayesian optimization (Snoek et al., 2012) with a total time budget of up to 3,600 seconds. In particular, ensembles combine $K = 10$ base rules obtained by 360 seconds of automated model selection. The 'Baseline' method corresponds to standard SGD training; 'AdaBoost' for NNs is implemented as in Wang et al. (2021) for image datasets and as in Schwenk & Bengio (2000) for tabular datasets; 'Ensemble' is implemented using randomly initialized models (Lakshminarayanan et al., 2017) for image datasets and using bagging (Breiman, 1996) for tabular datasets; and 'MBoost' is implemented as described in Section 5 using $n_{\text{held}} = 500$ for image datasets and $n_{\text{held}} = 100$ for tabular datasets. Comparisons with the methods in Cortes et al. (2014) tailored for deep decision trees are shown in Appendix F.2.

Table 1 shows the average classification error, expected calibration error (ECE), and running times (relative to the baseline) across methods using 10-fold cross-validation. MBoost improves overall performance in both accuracy and calibration, without additional computational overhead (the baseline is faster in the tabular datasets because the automated model selection lacks warm starts).

We further analyze the improved performance of MBoost with respect to current methods by assessing the error over boosting rounds. Figure 2 shows the classification error across 100 rounds of boosting (one epoch per round), averaged over 10 random repetitions. The results illustrate the evolution of errors using the datasets CIFAR-10 'cat vs dog' and 'real vs fake face'. While strong rules already provide good performance in the initial rounds, MBoost continues to improve across rounds, whereas the conventional methods tend to overfit as training progresses.

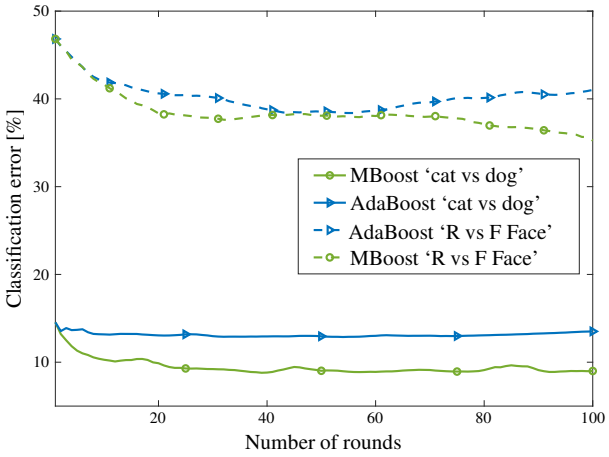


Figure 2: Classification error across boosting rounds.

The experiments in this section and in Appendix F show that MBoost can reliably combine NNs, outperforming current boosting approaches. The results show that the proposed methods provide improved classification errors and uncertainty assessments.

7 Conclusion

The paper presents a minimax approach for boosting (MBoost) that can leverage strong rules, including neural networks. Unlike existing techniques, MBoost minimizes the worst-case expected loss and accounts for the generalization behavior of individual base rules. The theoretical guarantees show that MBoost can effectively avoid overfitting and achieve performance close to the Bayes risk. The paper also introduces efficient learning algorithms that use data-driven estimates of generalization gaps. Both theoretical and experimental results show that MBoost can reliably combine strong rules, improving performance in terms of classification accuracy and uncertainty estimation. Overall, the boosting methodology introduced can help expand the success of boosting methods to more general scenarios that require the use of strong rules.

Limitations and future work: The main operational difference between MBoost and existing boosting approaches is the need to estimate the generalization gaps of the selected base rules. As described in the paper, this estimation can be effectively carried out by taking advantage of the standard per-round subsampling in boosting, at the expense of slightly reducing the sample size available to the base-learner. Another limitation is the restriction of the current formulation to problems of binary classification. This is common in the boosting literature, as the multiclass setting is significantly more cumbersome and technically involved, and is typically addressed as a subsequent extension. Accordingly, we plan to extend MBoost to multiclass problems in future work.

AI USE STATEMENT

In this work, we used generative AI tools for polishing the language of the manuscript, finding typos, and correcting grammatical errors. We have not used generative AI tools for developing the MBoost formulation, deriving the theoretical results or their proofs, designing or implementing the proposed methods, writing or running experimental code, or analyzing and interpreting results. Tasks involving data collection or human participants are not applicable to this work. We have reviewed all AI-assisted work: all suggested language edits were checked by the authors to ensure that they did not alter the technical content, results, or claims of the paper. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work proposes a minimax boosting methodology (MBoost) for combining strong base rules such as neural networks. All experiments use 18 publicly available benchmark datasets obtained from the UCI repository, TensorFlow Datasets, and Kaggle (Appendix F.1, Table 2), together with synthetic data (Appendix F.3). Some of these datasets contain medical images (diabetic retinopathy, chest X-ray, and breast histopathology) or face images (real and fake face detection); we use them only as released by their providers for research purposes, do not collect any new data, and make no attempt to identify individuals. The work involves no human subjects and releases no new datasets. We do not foresee harmful applications specific to the proposed method beyond those common to general-purpose classification techniques, and any deployment in sensitive domains such as medical diagnosis would require separate validation.

REPRODUCIBILITY STATEMENT

We have made several efforts to ensure the reproducibility of our results. For the theoretical results, all assumptions are stated in Theorems 1–3, and complete proofs are provided in Appendices B–D, building on the strong duality lemma in Appendix A. The learning algorithm is fully specified in Algorithm 1 (Section 5), and the derivation of its Gauss–Southwell coordinate descent updates is given in Appendix E. For the experiments, Appendix F.1 details the datasets and their binary reductions, preprocessing steps, network architectures and training hyperparameters, the number of held-out samples used to estimate generalization gaps, the 10-fold stratified cross-validation protocol, the computation of ECE, and the hardware used; the main characteristics of all datasets are summarized in Table 2. The generation process for the synthetic data is described in Appendix F.3, and the implementation of the Deep Boosting comparison in Appendix F.2 is based on the authors' public code. The implementation code for MBoost and all compared methods is included in the supplementary material.

REFERENCES

Charalambos D. Aliprantis and Kim C. Border. *Infinite dimensional analysis*. Springer-Verlag, Berlin, 1994.

Akshay Balsubramani and Yoav Freund. Optimal binary classifier aggregation for general losses. In *Advances in Neural Information Processing Systems*, volume 29, pp. 5032–5039, 2016.

Peter L. Bartlett and Mikhail Traskin. AdaBoost is consistent. *Journal of Machine Learning Research*, 8(78):2347–2368, 2007.

Jonathan M. Borwein and Qiji J. Zhu. *Techniques of variational analysis*. Springer, Berlin, 2004.

Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

Tianqi Chen and Carlos Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794, 2016.

Corinna Cortes, Mehryar Mohri, and Umar Syed. Deep boosting. In *Proceedings of the 31st International Conference on Machine Learning*, pp. 1179–1187, 2014.

Corinna Cortes, Mehryar Mohri, and Dmitry Storcheus. Regularized gradient boosting. In *Advances in Neural Information Processing Systems*, volume 32, pp. 5449–5458, 2019.

Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in Neural Information Processing Systems*, volume 27, pp. 1646–1654, 2014.

Ronald DeVore, Boris Hanin, and Guergana Petrova. Neural network approximation. *Acta Numerica*, 30:327–444, 2021.

Dheeru Dua and Casey Graff. UCI Machine Learning Repository, 2017. URL `http://archive.ics.uci.edu/ml`.

John Duchi and Hongseok Namkoong. Variance-based regularization with convex objectives. *Journal of Machine Learning Research*, 20(68):1–55, 2019.

John Duchi and Yoram Singer. Boosting with structural sparsity. In *26th International Conference on Machine Learning*, pp. 297–304, 2009.

Yoav Freund and Robert E. Schapire. Experiments with a new boosting algorithm. In *Proceedings of the 13th International Conference on Machine Learning*, pp. 148–156, 1996.

Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5):1189–1232, 2001.

Jerome H. Friedman. Stochastic gradient boosting. *Computational Statistics & Data Analysis*, 38 (4):367–378, 2002.

Leo Grinsztajn, Edouard Oyallon, and Gael Varoquaux. Why do tree-based models still outperform deep learning on typical tabular data? In *Advances in Neural Information Processing Systems*, volume 35, pp. 507–520, 2022.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, New York, NY, 2nd edition, 2009.

Tom Hope, Oscar Chang, Ehsan Amid, Yoni Halpern, Zachary Nado, Douglas Eck, and Rif A. Saurous. TensorFlow Datasets: A collection of ready-to-use datasets. `https://www.tensorflow.org/datasets`, 2022.

Gao Huang, Yixuan Li, Geoff Pleiss, Zhuang Liu, John E. Hopcroft, and Kilian Q. Weinberger. Snapshot ensembles: Train 1, get m for free. In *International Conference on Learning Representations*, 2017.

Sai Praneeth Karimireddy, Anastasia Koloskova, Sebastian U. Stich, and Martin Jaggi. Efficient greedy coordinate descent for composite problems. In *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics*, pp. 2887–2896, 2019.

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. LightGBM: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems*, volume 30, pp. 3146–3154, 2017.

Hidetoshi Komiya. Elementary proof for Sion's minimax theorem. *Kodai Mathematical Journal*, 11(1):5–7, 1988.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, pp. 6402–6413, 2017.

Jaeho Lee and Maxim Raginsky. Minimax statistical learning with Wasserstein distances. In *Advances in Neural Information Processing Systems*, volume 31, pp. 2692–2701, 2018.

Gábor Lugosi and Nicolas Vayatis. On the Bayes-risk consistency of regularized boosting methods. *The Annals of Statistics*, 32(1):30–55, 2004.

Santiago Mazuelas and Verónica Alvarez. Robust minimax boosting with performance guarantees. In *Advances in Neural Information Processing Systems*, volume 38, pp. 124827–124858, 2025.

Santiago Mazuelas, Andrea Zanoni, and Aritz Pérez. Minimax classification with 0-1 loss and performance guarantees. In *Advances in Neural Information Processing Systems*, volume 33, pp. 302–312, 2020.

Santiago Mazuelas, Yuan Shen, and Aritz Pérez. Generalized maximum entropy for supervised classification. *IEEE Transactions on Information Theory*, 68(4):2530–2550, April 2022.

Santiago Mazuelas, Mauricio Romero, and Peter Grünwald. Minimax risk classifiers with 0-1 loss. *Journal of Machine Learning Research*, 24(208):1–48, 2023.

H. N. Mhaskar and T. Poggio. Deep vs. shallow networks: An approximation theory perspective. *Analysis and Applications*, 14(06):829–848, 2016.

Mohammad Moghimi, Serge Belongie, Mohammad Saberian, Jian Yang, Nuno Vasconcelos, and Li-Jia Li. Boosted convolutional neural networks. In *Proceedings of the British Machine Vision Conference 2016*, 2016.

Giung Nam, Jongmin Yoon, Yoonho Lee, and Juho Lee. Diversity matters when learning from ensembles. In *Advances in Neural Information Processing Systems*, pp. 8367–8377, 2021.

Julie Nutini, Mark Schmidt, Issam Laradji, Michael Friedlander, and Hoyt Koepke. Coordinate descent converges faster with the Gauss-Southwell rule than random selection. In *Proceedings of the 32nd International Conference on Machine Learning*, pp. 1632–1641, 2015.

David Opitz and Richard Maclin. Popular ensemble methods: An empirical study. *Journal of Artificial Intelligence Research*, 11(1):169–198, July 1999.

Luis A. Ortega, Rafael Cabañas, and Andres Masegosa. Diversity and generalization in neural network ensembles. In *Proceedings of the 25th International Conference on Artificial Intelligence and Statistics*, pp. 11720–11743, 2022.

Matteo Pagliardini, Martin Jaggi, François Fleuret, and Sai Praneeth Karimireddy. Agree to disagree: Diversity through disagreement for better transferability. In *International Conference on Learning Representations*, 2023.

Sai Saketh Rambhatla, Michael J. Jones, and Rama Chellappa. An empirical analysis of boosting deep networks. In *International Joint Conference on Neural Networks*, pp. 1–9, 2022.

Robert E. Schapire. The strength of weak learnability. In *30th Annual Symposium on Foundations of Computer Science*, pp. 28–33, 1989.

Robert E. Schapire and Yoav Freund. *Boosting: Foundations and algorithms*. MIT Press, Cambridge, MA, 2012.

Holger Schwenk and Yoshua Bengio. Boosting neural networks. *Neural Computation*, 12(8):1869–1887, August 2000.

Scikit-learn library. Gradient boosting out-of-bag estimates. `https://scikit-learn.org/stable/auto_examples/ensemble/plot_gradient_boosting_oob.html`, 2025.

Soroosh Shafieezadeh-Abadeh, Daniel Kuhn, and Peyman Mohajerin Esfahani. Regularization via mass transportation. *Journal of Machine Learning Research*, 20(103):1–68, 2019.

Chunhua Shen, Hanxi Li, and Anton van den Hengel. Fully corrective boosting with arbitrary loss and regularization. *Neural Networks*, 48:44–58, 2013.

Zuowei Shen, Haizhao Yang, and Shijun Zhang. Nonlinear approximation via compositions. *Neural Networks*, 119:74–84, 2019.

Jasper Snoek, Hugo Larochelle, and Ryan P. Adams. Practical Bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems*, volume 25, 2012.

Feng Wang, Guoyizhe Wei, Qiao Liu, Jinxiang Ou, Xian Wei, and Hairong Lv. Boost neural networks by checkpoints. In *Advances in Neural Information Processing Systems*, pp. 19719–19729, 2021.

Manfred K. Warmuth, Jun Liao, and Gunnar Rätsch. Totally corrective boosting algorithms that maximize the margin. In *Proceedings of the 23rd International Conference on Machine Learning*, pp. 1001–1008, 2006.

APPENDICES

A STRONG DUALITY LEMMA

Some of the proofs for the results in the paper make use of Fenchel duality for linear optimization problems over probability measures. The next lemma provides the duality result needed for such proofs.

Lemma 4. Let $\mathcal{V}$ be an uncertainty set given by equation 6. For any $\mathrm{h} \in \mathrm{T}(\mathcal{X}, \mathcal{Y})$, we have

$$\sup_{\mathrm{p}\in\mathcal{V}} \mathbb{E}_{\mathrm{p}}\{\ell(\mathrm{h},(x,y))\} \leq \inf_{\boldsymbol{\mu}} \ \mathbf{e}^\top \boldsymbol{\mu} - \mathbb{E}_{\mathrm{p}_x^*}\{\min_{y\in\mathcal{Y}}(\mathbf{er}(x,y)^\top \boldsymbol{\mu} + \log \mathrm{h}(y|x))\} + \boldsymbol{\lambda}^\top |\boldsymbol{\mu}| \tag{23}$$

where $\mathbf{er}(x,y)$ is given by $\mathbf{er}(x,y) = \big[\mathbb{I}\{\hbar_1(x) \neq y\}, \mathbb{I}\{\hbar_2(x) \neq y\}, \ldots, \mathbb{I}\{\hbar_T(x) \neq y\}\big]^\top$, and $\mathbf{e}$ is given by equation 7. In addition, if $\mathcal{V}$ has non-empty interior, the 'sup' and 'inf' in equation 23 are attained and we have

$$\max_{\mathrm{p}\in\mathcal{V}} \mathbb{E}_{\mathrm{p}}\{\ell(\mathrm{h},(x,y))\} = \min_{\boldsymbol{\mu}} \ \mathbf{e}^\top \boldsymbol{\mu} - \mathbb{E}_{\mathrm{p}_x^*}\{\min_{y\in\mathcal{Y}}(\mathbf{er}(x,y)^\top \boldsymbol{\mu} + \log \mathrm{h}(y|x))\} + \boldsymbol{\lambda}^\top |\boldsymbol{\mu}|. \tag{24}$$

Proof. In the first step of the proof, we show that the right-hand-side in equation 23 is the Fenchel dual of the left-hand-side. Then, the result is obtained by showing that strong duality is satisfied whenever the uncertainty set has non-empty interior.

Let $M(\mathcal{X}\times\mathcal{Y})$ be the set of signed Borel measures over $\mathcal{X}\times\mathcal{Y}$ with bounded variation. Such a set is a Banach space with the total variation norm (see e.g., Chapter 10 in Aliprantis & Border (1994)).

If A is the linear mapping

$$\begin{array}{rccc} A: & M(\mathcal{X}\times\mathcal{Y}) & \to & \mathbb{R}^T \times \mathbb{R}^T \\ & \mathrm{p} & \mapsto & \big(\int \mathbf{er}(x,y) d\mathrm{p}(x,y), -\int \mathbf{er}(x,y) d\mathrm{p}(x,y)\big). \end{array} \tag{25}$$

It is clear that A is bounded and its adjoint operator is

$$\begin{array}{rccc} A^*: & \mathbb{R}^T \times \mathbb{R}^T & \to & F(\mathcal{X}\times\mathcal{Y}) \subseteq (M(\mathcal{X}\times\mathcal{Y}))^* \\ & (\boldsymbol{\mu}_1, \boldsymbol{\mu}_2) & \mapsto & \mathbf{er}(\cdot,\cdot)^\top (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) \end{array} \tag{26}$$

where $F(\mathcal{X}\times\mathcal{Y})$ is the set of bounded Borel measurable functions over $\mathcal{X}\times\mathcal{Y}$.

Then, we have

$$\sup_{\mathrm{p}\in\mathcal{V}} \mathbb{E}_{\mathrm{p}}\{\ell(\mathrm{h},(x,y))\} = -\inf_{\mathrm{p}\in M(\mathcal{X}\times\mathcal{Y})} f(\mathrm{p}) + g(A(\mathrm{p})) \tag{27}$$

where f and g are the lower semi-continuous convex functions

$$\begin{array}{rccc} g: & \mathbb{R}^T \times \mathbb{R}^T & \to & \mathbb{R}\cup\{\infty\} \\ & (\mathbf{v}_1, \mathbf{v}_2) & \mapsto & \left\{ \begin{array}{cc} 0 & \text{if } \mathbf{v}_1 \preceq \mathbf{e}+\boldsymbol{\lambda},\ \mathbf{v}_2 \preceq -\mathbf{e}+\boldsymbol{\lambda} \\ \infty & \text{otherwise} \end{array} \right. \end{array} \tag{28}$$

$$\begin{array}{rccc} f: & M(\mathcal{X}\times\mathcal{Y}) & \to & \mathbb{R}\cup\{\infty\} \\ & \mathrm{p} & \mapsto & \left\{ \begin{array}{cc} \int \log \mathrm{h}(y|x) d\mathrm{p}(x,y) & \text{if } \mathrm{p} \succeq 0 \text{ and } \sum_{y\in\mathcal{Y}} \mathrm{p}(x,y) = \mathrm{p}_x^*(x),\ \forall x \in \mathcal{X} \\ \infty & \text{otherwise.} \end{array} \right. \end{array} \tag{29}$$

Then, the Fenchel dual (see e.g., Borwein & Zhu (2004)) of the left hand side in equation 23 is

$$\inf_{\boldsymbol{\mu}_1,\boldsymbol{\mu}_2} f^*(A^*(\boldsymbol{\mu}_1,\boldsymbol{\mu}_2)) + g^*(-\boldsymbol{\mu}_1, -\boldsymbol{\mu}_2) \tag{30}$$

where f^* and g^* are the conjugate functions of f and g. If $w \in F(\mathcal{X}\times\mathcal{Y})$, we have

$$\begin{aligned} f^*(w) &= \sup_{\mathrm{p}(y|x)} \int \Big(\sum_{y\in\mathcal{Y}} (w(x,y) - \log \mathrm{h}(y|x)) \mathrm{p}(y|x) \Big) d\mathrm{p}_x^*(x) \\ &= \mathbb{E}_{\mathrm{p}_x^*}\big\{ \max_{y\in\mathcal{Y}} (w(x,y) - \log \mathrm{h}(y|x)) \big\} \end{aligned}$$

and $g^*(-\boldsymbol{\mu}_1, -\boldsymbol{\mu}_2)$ is given by

$$\begin{aligned} &\sup \quad -\mathbf{v}_1^\top \boldsymbol{\mu}_1 - \mathbf{v}_2^\top \boldsymbol{\mu}_2 \\ &\text{s.t.} \quad \mathbf{v}_1 \preceq \mathbf{e} + \boldsymbol{\lambda},\ \mathbf{v}_2 \preceq -\mathbf{e} + \boldsymbol{\lambda} \\ &\qquad = \begin{cases} -(\mathbf{e}+\boldsymbol{\lambda})^\top \boldsymbol{\mu}_1 - (-\mathbf{e}+\boldsymbol{\lambda})^\top \boldsymbol{\mu}_2 & \text{if} \quad -\boldsymbol{\mu}_1 \succeq \mathbf{0}, -\boldsymbol{\mu}_2 \succeq \mathbf{0} \\ \infty & \text{otherwise.} \end{cases} \end{aligned}$$

Hence, the dual problem (30) becomes

$$\inf_{\boldsymbol{\mu}_1, \boldsymbol{\mu}_2 \succeq \mathbf{0}} (\mathbf{e}+\boldsymbol{\lambda})^\top \boldsymbol{\mu}_1 + (-\mathbf{e}+\boldsymbol{\lambda})^\top \boldsymbol{\mu}_2 + \mathbb{E}_{\mathrm{p}_x^*}\big\{\max_{y \in \mathcal{Y}} (\mathbf{er}(x,y)^\top (\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1) - \log \mathrm{h}(y|x))\big\} \tag{31}$$

$$= \inf_{\boldsymbol{\mu}} \mathbf{e}^\top \boldsymbol{\mu} - \mathbb{E}_{\mathrm{p}_x^*}\big\{\min_{y \in \mathcal{Y}} (\mathbf{er}(x,y)^\top \boldsymbol{\mu} + \log \mathrm{h}(y|x))\big\} + \boldsymbol{\lambda}^\top |\boldsymbol{\mu}|. \tag{32}$$

The expression in equation 32 leading to the weak-duality result in equation 23 is obtained by taking $\boldsymbol{\mu} = \boldsymbol{\mu}_1 - \boldsymbol{\mu}_2$. Firstly, we can consider only pairs $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2$ such that $\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2 = |\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2|$ because for any pair $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2$, we have that $\tilde{\boldsymbol{\mu}}_1 = (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)_+$, $\tilde{\boldsymbol{\mu}}_2 = (\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)_+$ satisfies $\tilde{\boldsymbol{\mu}}_1 - \tilde{\boldsymbol{\mu}}_2 = \boldsymbol{\mu}_1 - \boldsymbol{\mu}_2$, and $\boldsymbol{\lambda}^\top |\tilde{\boldsymbol{\mu}}_1 - \tilde{\boldsymbol{\mu}}_2| = \boldsymbol{\lambda}^\top (\tilde{\boldsymbol{\mu}}_1 + \tilde{\boldsymbol{\mu}}_2) \leq \boldsymbol{\lambda}^\top (\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2)$.

The equality in equation 24 is obtained by showing that strong duality holds since $\mathbf{0} \in \mathrm{int}(\mathrm{dom}\, g - A\mathrm{dom}\, f)$ (see e.g., Chapter 4 in Borwein & Zhu (2004)), where 'dom' denotes the set where an extended-valued function takes finite values, and 'int' denotes the interior of a set.

If the interior of $\mathcal{V}$ is not empty, there exists $\mathrm{p} \in \mathcal{V}$ with $|\mathbb{E}_{\mathrm{p}}\{\mathbf{er}(x,y)\} - \mathbf{e}| \prec \boldsymbol{\lambda}$, and hence there exists an $\varepsilon > 0$ such that $\varepsilon < \lambda_k - |\mathbb{E}_{\mathrm{p}}\{\mathrm{er}_k(x,y)\} - e_k|$ for $k = 1, 2, \ldots, T$. Then, we obtain the strong duality result by showing that the ball with radius ε centered at $\mathbf{0} \in \mathbb{R}^{2T}$, $B(\mathbf{0}, \varepsilon)$ satisfies $B(\mathbf{0}, \varepsilon) \subset (\mathrm{dom}\, g - A\mathrm{dom}\, f) \subset \mathbb{R}^{2T}$. Such an inclusion can be easily shown because for any $\mathbf{z} \in B(\mathbf{0}, \varepsilon) \subset \mathbb{R}^{2T}$, we have

$$\begin{aligned} \big(z_1, z_2, \ldots, z_T\big) &= \mathbb{E}_{\mathrm{p}}\{\mathbf{er}(x,y)\} + \big(z_1, z_2, \ldots, z_T\big) - \mathbb{E}_{\mathrm{p}}\{\mathbf{er}(x,y)\} \\ \big(z_{T+1}, z_{T+2}, \ldots, z_{2T}\big) &= -\mathbb{E}_{\mathrm{p}}\{\mathbf{er}(x,y)\} + \big(z_{T+1}, z_{T+2}, \ldots, z_{2T}\big) + \mathbb{E}_{\mathrm{p}}\{\mathbf{er}(x,y)\} \end{aligned}$$

with $(\mathbb{E}_{\mathrm{p}}\{\mathbf{er}(x,y)\}, -\mathbb{E}_{\mathrm{p}}\{\mathbf{er}(x,y)\}) \in A\mathrm{dom} f$ and

$$\big(\mathbb{E}_{\mathrm{p}}\{\mathbf{er}(x,y)\} + \big(z_1, z_2, \ldots, z_T\big), -\mathbb{E}_{\mathrm{p}}\{\mathbf{er}(x,y)\} + \big(z_{T+1}, z_{T+2}, \ldots, z_{2T}\big) \in \mathrm{dom}\, g$$

by definition of ε. Finally, since strong duality holds and $\mathcal{V}$ is not empty we have that the optimal value in (24) is finite and hence the optimum of the dual is attained (Borwein & Zhu, 2004), so that the 'inf' in equation 32 becomes 'min' in equation 24. Moreover, the 'sup' in equation 27 is actually a 'max' as in equation 24 because the set $\mathcal{V}$ is weakly* compact due to Prokhorov's theorem (see e.g., Section 15.5 in Aliprantis & Border (1994)) since the distributions in $\mathcal{V}$ share the same marginal over the Borel set $\mathcal{X}$. □

B Proof of Theorem 1

Using Lemma 4, the minimax problem in (5) is equivalent to

$$\min_{\boldsymbol{\mu}} \mathbf{e}^\top \boldsymbol{\mu} + \boldsymbol{\lambda}^\top |\boldsymbol{\mu}| - \sup_{\mathrm{h} \in \mathrm{T}(\mathcal{X}, \mathcal{Y})} \mathbb{E}_{\mathrm{p}_x^*}\big\{\min_{y \in \mathcal{Y}} (\mathbf{er}(x,y)^\top \boldsymbol{\mu} + \log \mathrm{h}(y|x))\big\} \tag{33}$$

for $\mathbf{er}(x,y) = \big[\mathbb{I}\{\hbar_1(x) \neq y\}, \mathbb{I}\{\hbar_2(x) \neq y\}, \ldots, \mathbb{I}\{\hbar_T(x) \neq y\}\big]^\top$. If, for each $\mathrm{h} \in \mathrm{T}(\mathcal{X}, \mathcal{Y})$ and $x \in \mathcal{X}$, $\nu_{\mathrm{h}}(x)$ denotes the value

$$\nu_{\mathrm{h}}(x) = \min_{y \in \mathcal{Y}} (\mathbf{er}(x,y)^\top \boldsymbol{\mu} + \log \mathrm{h}(y|x))$$

we have

$$\begin{aligned} \nu_{\mathrm{h}}(x) &\leq \mathbf{er}(x,y)^\top \boldsymbol{\mu} + \log \mathrm{h}(y|x),\ \forall x \in \mathcal{X}, y \in \mathcal{Y} \\ &\Rightarrow \mathrm{h}(y|x) \geq \exp\{\nu_{\mathrm{h}}(x) - \mathbf{er}(x,y)^\top \boldsymbol{\mu}\},\ \forall x \in \mathcal{X}, y \in \mathcal{Y} \\ &\Rightarrow \sum_{y \in \mathcal{Y}} \exp\{\nu_{\mathrm{h}}(x) - \mathbf{er}(x,y)^\top \boldsymbol{\mu}\} \leq 1,\ \forall x \in \mathcal{X} \\ &\Rightarrow \nu_{\mathrm{h}}(x) \leq -\log \sum_{y \in \mathcal{Y}} \exp\{-\mathbf{er}(x,y)^\top \boldsymbol{\mu}\},\ \forall x \in \mathcal{X}. \end{aligned}$$

Then, it is clear that the rule $\mathrm{h}^{\boldsymbol{\mu}}(y|x) = (1 + \exp(-y\boldsymbol{\hbar}(x)^\top \boldsymbol{\mu}))^{-1}$ is a solution of

$$\sup_{\mathrm{h}\in\mathsf{T}(\mathcal{X},\mathcal{Y})} \mathbb{E}_{\mathrm{p}^*_x}\big\{\min_{y\in\mathcal{Y}}(\mathbf{er}(x,y)^\top\boldsymbol{\mu} + \log \mathrm{h}(y|x))\big\} = \sup_{\mathrm{h}\in\mathsf{T}(\mathcal{X},\mathcal{Y})} \mathbb{E}_{\mathrm{p}^*_x}\{\nu_{\mathrm{h}}(x)\}$$

since, for any $x \in \mathcal{X}$ and $\mathrm{h} \in \mathsf{T}(\mathcal{X},\mathcal{Y})$, we have

$$\nu_{\mathrm{h}^{\boldsymbol{\mu}}}(x) = -\log \sum_{y\in\mathcal{Y}} \exp\{-\mathbf{er}(x,y)^\top\boldsymbol{\mu}\} \geq \nu_{\mathrm{h}}(x)$$

because

$$\begin{aligned}
&\mathbf{er}(x,y)^\top\boldsymbol{\mu} + \log \mathrm{h}^{\boldsymbol{\mu}}(y|x) \\
&\quad= \Big(\frac{1-y\boldsymbol{\hbar}(x)}{2}\Big)^\top\boldsymbol{\mu} + \log \frac{1}{1+\exp\{-y\boldsymbol{\hbar}(x)^\top\boldsymbol{\mu}\}} \\
&\quad= \Big(\frac{1-y\boldsymbol{\hbar}(x)}{2}\Big)^\top\boldsymbol{\mu} + \log \frac{\exp\{-\big(\frac{1-y\boldsymbol{\hbar}(x)}{2}\big)^\top\boldsymbol{\mu}\}}{\exp\{-\big(\frac{1-y\boldsymbol{\hbar}(x)}{2}\big)^\top\boldsymbol{\mu}\} + \exp\{-\big(\frac{1+y\boldsymbol{\hbar}(x)}{2}\big)^\top\boldsymbol{\mu}\}} \\
&\quad= -\log \sum_{y\in\mathcal{Y}} \exp\{-\mathbf{er}(x,y)^\top\boldsymbol{\mu}\} \\
&\quad\Rightarrow \nu_{\mathrm{h}^{\boldsymbol{\mu}}}(x) = -\log \sum_{y\in\mathcal{Y}} \exp\{-\mathbf{er}(x,y)^\top\boldsymbol{\mu}\}.
\end{aligned}$$

Therefore, we directly obtain the first part of the theorem since

$$\log \psi(x,\boldsymbol{\mu}) = \log \sum_{y\in\mathcal{Y}} \exp\{-\mathbf{er}(x,y)^\top\boldsymbol{\mu}\}.$$

For the second part of the theorem, we first show that

$$\min_{\mathrm{h}\in\mathsf{T}(\mathcal{X},\mathcal{Y})} \max_{\mathrm{p}\in\mathcal{V}} \mathbb{E}_{\mathrm{p}}\{\ell(\mathrm{h},(x,y))\} = \max_{\mathrm{p}\in\mathcal{V}} \min_{\mathrm{h}\in\mathsf{T}(\mathcal{X},\mathcal{Y})} \mathbb{E}_{\mathrm{p}}\{\ell(\mathrm{h},(x,y))\} \tag{34}$$

as a consequence of Sion's minimax theorem (see e.g., Komiya (1988)). The assumptions of that theorem are satisfied because: 1) the set $\mathsf{T}(\mathcal{X},\mathcal{Y})$ is convex and compact in the weak* topology, convexity follows from pointwise mixing and compactness from Tychonoff's theorem (see e.g., Aliprantis & Border (1994)) since $\Delta(\mathcal{Y}) \simeq [0,1]$ is compact; 2) the set $\mathcal{V}$ is convex and compact in the weak* topology, convexity follows from linearity of $\mathrm{err}(\mathrm{p},\hbar_k)$ in p, and compactness from Prokhorov's theorem (see e.g., Section 15.5 in Aliprantis & Border (1994)) since all distributions in $\mathcal{V}$ share the fixed marginal p^*_x; and 3) the function $\mathbb{E}_{\mathrm{p}}\{\ell(\mathrm{h},(x,y))\}$ is continuous and concave in p by dominated convergence under weak* convergence, and lower semicontinuous and convex in h since the log loss $\ell(\mathrm{h},(x,y)) = -\log \mathrm{h}(y|x)$ is lower semicontinuous and convex at each x, and the expectation preserves both properties.

There exists a worst-case distribution $\hat{\mathrm{p}} \in \mathcal{V}$ because the equality in equation 34 implies that

$$\max_{\mathrm{p}\in\mathcal{V}} \min_{\mathrm{h}\in\mathsf{T}(\mathcal{X},\mathcal{Y})} \mathbb{E}_{\mathrm{p}}\{\ell(\mathrm{h},(x,y))\} = \max_{\mathrm{p}\in\mathcal{V}} \mathbb{E}_{\mathrm{p}}\{\ell(\mathrm{h}^{\boldsymbol{\mu}_{\boldsymbol{\lambda}}},(x,y))\} \tag{35}$$

since $\mathrm{h}^{\boldsymbol{\mu}_{\boldsymbol{\lambda}}}$ is a solution of the left hand side of equation 34. Then, as a consequence of Lemma 4, the solution of optimization problem (35) is attained by some $\hat{\mathrm{p}} \in \mathcal{V}$, and using again the equality in equation 34 we have

$$\mathrm{h}^{\boldsymbol{\mu}_{\boldsymbol{\lambda}}} \in \arg\min_{\mathrm{h}\in\mathsf{T}(\mathcal{X},\mathcal{Y})} \mathbb{E}_{\hat{\mathrm{p}}}\{\ell(\mathrm{h},(x,y))\} = \arg\min_{\mathrm{h}\in\mathsf{T}(\mathcal{X},\mathcal{Y})} \mathbb{E}_{\hat{\mathrm{p}}}\{-\log(\mathrm{h}(y|x))\}.$$

Therefore, we obtain that $\mathrm{h}^{\boldsymbol{\mu}_{\boldsymbol{\lambda}}}(y|x) = \hat{\mathrm{p}}(y|x)$ ($\mathrm{p}^*_x(x)$-almost surely) because the log-loss is strictly proper, which directly gives that $\mathrm{p}^{\boldsymbol{\mu}_{\boldsymbol{\lambda}}}$ is the worst-case distribution in $\mathcal{V}$.

C PROOF OF THEOREM 2

In the proof we denote by $\mathcal{V}_{\boldsymbol{\lambda}}$ and $\mathcal{V}_{\boldsymbol{\lambda}^*}$ the uncertainty sets corresponding to generalization gap estimates $\boldsymbol{\lambda}$ and actual values $\boldsymbol{\lambda}^*$.

For the first part of the theorem, we have that the underlying distribution p^* is included in $\mathcal{V}_{\boldsymbol{\lambda}^*}$ directly using the definitions of $\boldsymbol{\lambda}^*$ and uncertainty set in equation 3 and equation 6. Then, using Lemma 4 we have

$$\begin{aligned} R(\mathrm{h}^{\boldsymbol{\mu}_{\boldsymbol{\lambda}}}) &= \mathbb{E}_{\mathrm{p}^*}\{\ell(\mathrm{h}^{\boldsymbol{\mu}_{\boldsymbol{\lambda}}}, (x,y))\} \leq \sup_{\mathrm{p}\in\mathcal{V}_{\boldsymbol{\lambda}^*}} \mathbb{E}_{\mathrm{p}}\{\ell(\mathrm{h}^{\boldsymbol{\mu}_{\boldsymbol{\lambda}}}, (x,y))\} \\ &\leq \inf_{\boldsymbol{\mu}}\ \mathbf{e}^\top\boldsymbol{\mu} - \mathbb{E}_{\mathrm{p}^*_x}\{\min_{y\in\mathcal{Y}}(\mathbf{er}(x,y)^\top\boldsymbol{\mu} + \log \mathrm{h}^{\boldsymbol{\mu}_{\boldsymbol{\lambda}}}(y|x))\} + (\boldsymbol{\lambda}^*)^\top|\boldsymbol{\mu}| \\ &\leq \mathbf{e}^\top\boldsymbol{\mu}_{\boldsymbol{\lambda}} - \mathbb{E}_{\mathrm{p}^*_x}\{\min_{y\in\mathcal{Y}}(\mathbf{er}(x,y)^\top\boldsymbol{\mu}_{\boldsymbol{\lambda}} + \log \mathrm{h}^{\boldsymbol{\mu}_{\boldsymbol{\lambda}}}(y|x))\} + (\boldsymbol{\lambda}^*)^\top|\boldsymbol{\mu}_{\boldsymbol{\lambda}}| \end{aligned} \tag{36}$$

where $\mathbf{er}(x,y) = [\mathbb{I}\{\hbar_1(x)\neq y\}, \mathbb{I}\{\hbar_2(x)\neq y\}, \ldots, \mathbb{I}\{\hbar_T(x)\neq y\}]^\top$. Hence, the inequality in equation 36 leads to that in equation 16 by adding and subtracting $\boldsymbol{\lambda}^\top|\boldsymbol{\mu}_{\boldsymbol{\lambda}}|$ and because for any $\boldsymbol{\mu}$ we have

$$\begin{aligned} &\mathbf{er}(x,y)^\top\boldsymbol{\mu} + \log \mathrm{h}^{\boldsymbol{\mu}}(y|x) \qquad (37)\\ &\quad= \Big(\frac{1-y\boldsymbol{\hbar}(x)}{2}\Big)^\top\boldsymbol{\mu} + \log\frac{1}{1+\exp\{-y\boldsymbol{\hbar}(x)^\top\boldsymbol{\mu}\}} \\ &\quad= \Big(\frac{1-y\boldsymbol{\hbar}(x)}{2}\Big)^\top\boldsymbol{\mu} + \log\frac{\exp\{-\big(\frac{1-y\boldsymbol{\hbar}(x)}{2}\big)^\top\boldsymbol{\mu}\}}{\exp\{-\big(\frac{1-y\boldsymbol{\hbar}(x)}{2}\big)^\top\boldsymbol{\mu}\} + \exp\{-\big(\frac{1+y\boldsymbol{\hbar}(x)}{2}\big)^\top\boldsymbol{\mu}\}} \\ &\quad= -\log\sum_{y\in\mathcal{Y}}\exp\{-\mathbf{er}(x,y)^\top\boldsymbol{\mu}\} = -\log(\psi(x,\boldsymbol{\mu})). \qquad (38) \end{aligned}$$

For the second part of the theorem, using Lemma 4 and equation 38 we have

$$\begin{aligned} R_{\boldsymbol{\lambda}^*} &= \inf_{\mathrm{h}\in\mathsf{T}(\mathcal{X},\mathcal{Y})}\ \sup_{\mathrm{p}\in\mathcal{V}} \mathbb{E}_{\mathrm{p}}\{\ell(\mathrm{h},(x,y))\} \\ &\leq \inf_{\boldsymbol{\mu}}\ \mathbf{e}^\top\boldsymbol{\mu} - \mathbb{E}_{\mathrm{p}^*_x}\{\min_{y\in\mathcal{Y}}(\mathbf{er}(x,y)^\top\boldsymbol{\mu} + \log \mathrm{h}^{\boldsymbol{\mu}}(y|x))\} + (\boldsymbol{\lambda}^*)^\top|\boldsymbol{\mu}| \\ &= \inf_{\boldsymbol{\mu}}\ \mathbf{e}^\top\boldsymbol{\mu} + \mathbb{E}_{\mathrm{p}^*_x}\{\log\psi(x,\boldsymbol{\mu})\} + (\boldsymbol{\lambda}^*)^\top|\boldsymbol{\mu}| \\ &\leq \mathbf{e}^\top\boldsymbol{\mu}_{\boldsymbol{\lambda}} + \mathbb{E}_{\mathrm{p}^*_x}\{\log\psi(x,\boldsymbol{\mu}_{\boldsymbol{\lambda}})\} + (\boldsymbol{\lambda}^*)^\top|\boldsymbol{\mu}_{\boldsymbol{\lambda}}| = R_{\boldsymbol{\lambda}} + (\boldsymbol{\lambda}^*-\boldsymbol{\lambda})^\top|\boldsymbol{\mu}_{\boldsymbol{\lambda}}|. \end{aligned} \tag{39}$$

In addition, if $\mathcal{V}_{\boldsymbol{\lambda}^*}$ has non-empty interior, using Theorem 1 we have

$$\begin{aligned} R_{\boldsymbol{\lambda}} &= \min_{\boldsymbol{\mu}}\ \mathbf{e}^\top\boldsymbol{\mu} + \mathbb{E}_{\mathrm{p}^*_x}\{\log\psi(x,\boldsymbol{\mu})\} + \boldsymbol{\lambda}^\top|\boldsymbol{\mu}| \\ &\leq \mathbf{e}^\top\boldsymbol{\mu}_{\boldsymbol{\lambda}^*} + \mathbb{E}_{\mathrm{p}^*_x}\{\log\psi(x,\boldsymbol{\mu}_{\boldsymbol{\lambda}^*})\} + (\boldsymbol{\lambda}^*)^\top|\boldsymbol{\mu}_{\boldsymbol{\lambda}^*}| + (\boldsymbol{\lambda}-\boldsymbol{\lambda}^*)^\top|\boldsymbol{\mu}_{\boldsymbol{\lambda}^*}| \qquad (40)\\ &= R_{\boldsymbol{\lambda}^*} + (\boldsymbol{\lambda}-\boldsymbol{\lambda}^*)^\top|\boldsymbol{\mu}_{\boldsymbol{\lambda}^*}|. \qquad (41) \end{aligned}$$

Then, we have

$$R_{\boldsymbol{\lambda}} - R_{\boldsymbol{\lambda}^*} \leq (\boldsymbol{\lambda}-\boldsymbol{\lambda}^*)^\top|\boldsymbol{\mu}_{\boldsymbol{\lambda}^*}|. \tag{42}$$

So that, the inequality in equation 18 is obtained from equation 42 adding in the left and right sides $(\boldsymbol{\lambda}^*-\boldsymbol{\lambda})^\top|\boldsymbol{\mu}_{\boldsymbol{\lambda}}|$ and using reverse triangle and Hölder's inequalities.

Finally, the third part of the theorem is obtained because

$$R(\mathrm{h}^{\boldsymbol{\mu}}) \leq \mathbf{e}^\top\boldsymbol{\mu} + \mathbb{E}_{\mathrm{p}^*_x}\{\log\psi(x,\boldsymbol{\mu})\} + (\boldsymbol{\lambda}^*)^\top|\boldsymbol{\mu}| \tag{43}$$

as a consequence of Lemma 4, equation 38, and the fact that $\mathrm{p}^*\in\mathcal{V}_{\boldsymbol{\lambda}^*}$. Therefore, the inequality in equation 20 follows by adding and subtracting $\boldsymbol{\lambda}^\top|\boldsymbol{\mu}|$ in the right hand side of equation 43, and using equation 19.

D PROOF OF THEOREM 3

The result is obtained by bounding $R(\mathrm{h}^{\boldsymbol{\mu}_{\mathrm{B}}}) - R_{\text{Bayes}}$ and $R(\mathrm{h}^{\boldsymbol{\mu}}) - R(\mathrm{h}^{\boldsymbol{\mu}_{\mathrm{B}}})$ and using the straightforward decomposition

$$R(\mathrm{h}^{\boldsymbol{\mu}}) = R_{\text{Bayes}} + R(\mathrm{h}^{\boldsymbol{\mu}_{\mathrm{B}}}) - R_{\text{Bayes}} + R(\mathrm{h}^{\boldsymbol{\mu}}) - R(\mathrm{h}^{\boldsymbol{\mu}_{\mathrm{B}}}). \tag{44}$$

For the first difference, we have

$$\begin{aligned} R(\mathrm{h}^{\boldsymbol{\mu}_{\mathrm{B}}}) - R_{\text{Bayes}} &= \mathbb{E}_{\mathrm{p}^*_x}\Big\{ \sum_{y\in\mathcal{Y}} \mathrm{p}^*(y|x)(\log \mathrm{p}^*(y|x) - \log(\sigma(y\boldsymbol{\hbar}(x)^\top \boldsymbol{\mu}_{\mathrm{B}}))\Big\} \\ &= \mathbb{E}_{\mathrm{p}^*_x}\big\{\mathrm{kl}\big(\mathrm{p}^*(1|x)||\sigma(\boldsymbol{\hbar}(x)^\top \boldsymbol{\mu}_{\mathrm{B}})\big)\big\} \leq \varepsilon_{\text{approx}} \end{aligned} \tag{45}$$

because the rule $\mathrm{p}^*(y|x)$ achieves the Bayes ℓ-risk since the log-loss is proper.

For the second difference, we have

$$\begin{aligned} &R(\mathrm{h}^{\boldsymbol{\mu}}) - R(\mathrm{h}^{\boldsymbol{\mu}_{\mathrm{B}}}) \\ &\quad = \mathbb{E}_{\mathrm{p}^*_x}\Big\{ \sum_{y\in\mathcal{Y}} \mathrm{p}^*(y|x) \log(\sigma(y\boldsymbol{\hbar}(x)^\top \boldsymbol{\mu}_{\mathrm{B}}))\Big\} - \mathbb{E}_{\mathrm{p}^*_x}\Big\{ \sum_{y\in\mathcal{Y}} \mathrm{p}^*(y|x) \log(\sigma(y\boldsymbol{\hbar}(x)^\top \boldsymbol{\mu}))\Big\} \end{aligned} \tag{46}$$

so using the fact that

$$\sigma(u) = \frac{\exp(u/2)}{\exp(u/2) + \exp(-u/2)} = \frac{\exp(-(1-u)/2)}{\exp(-(1-u)/2) + \exp(-(1+u)/2)} \tag{47}$$

we have that for any $\boldsymbol{\mu}$

$$\log(\sigma(y\boldsymbol{\hbar}(x)^\top \boldsymbol{\mu})) = -\log(\psi(x,\boldsymbol{\mu})) - \mathbf{er}(x,y)^\top \boldsymbol{\mu}$$

with $\mathbf{er}(x,y) = \big[\mathbb{I}\{\hbar_1(x) \neq y\}, \mathbb{I}\{\hbar_2(x) \neq y\}, \ldots, \mathbb{I}\{\hbar_T(x) \neq y\}\big]^\top$. Hence,

$$\begin{aligned} \mathbb{E}_{\mathrm{p}^*_x}\Big\{ \sum_{y\in\mathcal{Y}} \mathrm{p}^*(y|x) \log(\sigma(y\boldsymbol{\hbar}(x)^\top \boldsymbol{\mu}_{\mathrm{B}}))\Big\} &= -\mathbf{e}_{\text{out}}^\top \boldsymbol{\mu}_{\mathrm{B}} - \mathbb{E}_{\mathrm{p}^*_x}\{\log \psi(x, \boldsymbol{\mu}_{\mathrm{B}})\} \\ \mathbb{E}_{\mathrm{p}^*_x}\Big\{ \sum_{y\in\mathcal{Y}} \mathrm{p}^*(y|x) \log(\sigma(y\boldsymbol{\hbar}(x)^\top \boldsymbol{\mu}))\Big\} &= -\mathbf{e}_{\text{out}}^\top \boldsymbol{\mu} - \mathbb{E}_{\mathrm{p}^*_x}\{\log \psi(x, \boldsymbol{\mu})\} \end{aligned}$$

for $\mathbf{e}_{\text{out}} = [\mathrm{err}(\mathrm{p}^*, \hbar_1), \mathrm{err}(\mathrm{p}^*, \hbar_2), \ldots, \mathrm{err}(\mathrm{p}^*, \hbar_T)]^\top$. So that we get

$$R(\mathrm{h}^{\boldsymbol{\mu}}) - R(\mathrm{h}^{\boldsymbol{\mu}_{\mathrm{B}}}) = \mathbf{e}_{\text{out}}^\top \boldsymbol{\mu} - \mathbf{e}_{\text{out}}^\top \boldsymbol{\mu}_{\mathrm{B}} + \mathbb{E}_{\mathrm{p}^*_x}\{\log \psi(x, \boldsymbol{\mu})\} - \mathbb{E}_{\mathrm{p}^*_x}\{\log \psi(x, \boldsymbol{\mu}_{\mathrm{B}})\}. \tag{48}$$

Hence, adding and subtracting $\mathbf{e}^\top \boldsymbol{\mu}$ and $\boldsymbol{\lambda}^\top |\boldsymbol{\mu}|$, and using the ε_{opt} suboptimality of $\boldsymbol{\mu}$, we obtain the bound in equation 22 because

$$\begin{aligned} &R(\mathrm{h}^{\boldsymbol{\mu}}) - R(\mathrm{h}^{\boldsymbol{\mu}_{\mathrm{B}}}) \\ &\quad = \mathbf{e}^\top \boldsymbol{\mu} + \mathbb{E}_{\mathrm{p}^*_x}\{\log \psi(x, \boldsymbol{\mu})\} + \boldsymbol{\lambda}^\top |\boldsymbol{\mu}| + \mathbf{e}_{\text{out}}^\top (\boldsymbol{\mu} - \boldsymbol{\mu}_{\mathrm{B}}) - \mathbb{E}_{\mathrm{p}^*_x}\{\log \psi(x, \boldsymbol{\mu}_{\mathrm{B}})\} - \mathbf{e}^\top \boldsymbol{\mu} - \boldsymbol{\lambda}^\top |\boldsymbol{\mu}| \\ &\quad \leq \mathbf{e}^\top \boldsymbol{\mu}_{\mathrm{B}} + \boldsymbol{\lambda}^\top |\boldsymbol{\mu}_{\mathrm{B}}| + \varepsilon_{\text{opt}} + \mathbf{e}_{\text{out}}^\top (\boldsymbol{\mu} - \boldsymbol{\mu}_{\mathrm{B}}) - \mathbf{e}^\top \boldsymbol{\mu} - \boldsymbol{\lambda}^\top |\boldsymbol{\mu}| \\ &\quad = (\mathbf{e} - \mathbf{e}_{\text{out}})^\top \boldsymbol{\mu}_{\mathrm{B}} + \boldsymbol{\lambda}^\top |\boldsymbol{\mu}_{\mathrm{B}}| + (\mathbf{e}_{\text{out}} - \mathbf{e})^\top \boldsymbol{\mu} - \boldsymbol{\lambda}^\top |\boldsymbol{\mu}| + \varepsilon_{\text{opt}} \\ &\quad \leq (|\mathbf{e} - \mathbf{e}_{\text{out}}| + \boldsymbol{\lambda})^\top |\boldsymbol{\mu}_{\mathrm{B}}| + (|\mathbf{e} - \mathbf{e}_{\text{out}}| - \boldsymbol{\lambda})^\top |\boldsymbol{\mu}| + \varepsilon_{\text{opt}}. \end{aligned} \tag{49}$$

E DERIVATION OF GS COORDINATE DESCENT

The following describes how the expression for the weights in line 10 of Alg. 1 corresponds to approximate GS coordinate descent.

The absolute value of the descent gradient of (8) corresponding to a rule not previously selected (component k with $\mu_k = 0$) is given by

$$\begin{aligned} |\nabla_k^+| &= \big| \min_{u\in[0,1]} u\, \nabla_k (\mathbf{e}^\top \boldsymbol{\mu} + \mathbb{E}_{\mathrm{p}^*_x}\{\log \psi(x, \boldsymbol{\mu})\} + \boldsymbol{\lambda}^\top \boldsymbol{\mu})\big| \\ &= \big| \min(\mathrm{err}(\mathrm{p}_n, \hbar_k) - \mathrm{err}(\mathrm{p}^{\boldsymbol{\mu}}, \hbar_k) + \lambda_k, 0)\big| = \big(\mathrm{err}(\mathrm{p}^{\boldsymbol{\mu}}, \hbar_k) - \mathrm{err}(\mathrm{p}_n, \hbar_k) - \lambda_k\big)_+ \end{aligned} \tag{50}$$

as a direct consequence of equation 15 and the fact that we can consider only positive coefficients in (8). Then, taking into account that $\lambda_k \approx \lambda_k^*$, the descent gradient corresponding to base rules selected by fitting training samples can be approximated as

$$|\nabla_k^+| \approx \big(\mathrm{err}(\mathrm{p}^{\boldsymbol{\mu}}, \hbar_k) - \mathrm{err}(\mathrm{p}_n, \hbar_k) - \lambda_k^*\big)_+$$
$$= \big(\mathrm{err}(\mathrm{p}^{\boldsymbol{\mu}}, \hbar_k) - \mathrm{err}(\mathrm{p}^*, \hbar_k)\big)_+ = \frac{1}{2}\big(\mathbb{E}_{\mathrm{p}^*}\{y\hbar_k(x)\} - \mathbb{E}_{\mathrm{p}_x^*}\{\sum_{y\in\mathcal{Y}} \mathrm{h}^{\boldsymbol{\mu}}(y|x) y\hbar_k(x)\}\big)_+. \quad (51)$$

Hence, a base rule corresponding to an approximately maximum descent gradient can be found using a base-learner that fits weighted training samples $\{(x_i', y_i', w_i)\}_{i=1}^{n-n_{\text{held}}}$. Specifically, the expression for the weights in line 10 of Algorithm 1 is obtained because

$$\frac{1}{2}\big(\mathbb{E}_{\mathrm{p}^*}\{y\hbar_k(x)\} - \mathbb{E}_{\mathrm{p}_x^*}\{\sum_{y\in\mathcal{Y}} \mathrm{h}^{\boldsymbol{\mu}}(y|x) y\hbar_k(x)\}\big)$$
$$\approx \frac{1}{n-n_{\text{held}}}\sum_{i=1}^{n-n_{\text{held}}} \frac{y_i'\hbar_k(x_i')}{2} - \frac{1}{n-n_{\text{held}}}\sum_{i=1}^{n-n_{\text{held}}} \frac{\mathrm{h}^{\boldsymbol{\mu}}(y_i'|x_i')y_i'\hbar_k(x_i') + (1-\mathrm{h}^{\boldsymbol{\mu}}(y_i'|x_i'))(-y_i')\hbar_k(x_i')}{2}$$
$$= \frac{1}{n-n_{\text{held}}}\sum_{i=1}^{n-n_{\text{held}}} (1-\mathrm{h}^{\boldsymbol{\mu}}(y_i'|x_i'))y_i'\hbar_k(x_i'). \quad (52)$$

F Implementation details and additional experimental results

In the following we describe the datasets used in Section 6 and provide further implementation details. Then, we complement the results in the main paper in several aspects, including the usage of synthetic data and the assessment of estimates for generalization gaps.

In the first set of additional results, we further compare the performance and the running time of MBoost with benchmark methods using real-world datasets. In the second set of additional results, we evaluate the performance of MBoost using synthetic data. Finally, the third set of additional results assesses the impact of the estimates for the generalization gaps using both real-world and synthetic data.

F.1 Datasets and implementation details

We utilize 18 publicly available datasets that are widely used as benchmarks for supervised classification methods. The first nine datasets are formed by image data such as 'CIFAR-10', 'MNIST', 'FashionMNIST', 'Food101', and 'Real and fake face detection', as well as medical imaging tasks such as 'Diabetic retinopathy', 'Chest X-ray', and 'Breast histopathology'. For multiclass datasets, we perform a binary reduction as follows: in 'CIFAR-10', we extract two datasets corresponding to the categories 'cat vs dog' and the categories 'car vs truck'; in 'MNIST', we use the digits 4 and 9; in 'FashionMNIST', we distinguish between coat and pullover; in the 'Food101' dataset, we use pizza and sushi; and in 'Diabetic retinopathy', we distinguish between diabetic and non-diabetic patients. The remaining nine datasets are formed by tabular datasets known to be well-suited for NNs, covering diverse domains from biological ('Heart', 'Diabetes', 'Blood transfusion', 'Haberman'), financial ('German', 'Credit'), and physical ('Ion', 'Raisin', 'QSAR') signals. The datasets used can be found in the UCI repository (Dua & Graff, 2017), TensorFlow datasets (Hope et al., 2022), and the Kaggle website `https://www.kaggle.com`, and their main characteristics are summarized in Table 2.

Experiments were conducted on two compute nodes: (i) a GH200 node with a 72-core ARM Neoverse-V2 CPU, 512 GB RAM, and an NVIDIA GH200 GPU (480GB HBM3), and (ii) an A100X node with 48 CPU cores (2x Intel Xeon Gold 6336Y), 512GB RAM, and 2x NVIDIA A100X GPUs (80GB each).

MBoost is implemented as detailed in Algorithm 1. The estimates for generalization gaps are obtained using $n_{\text{held}} = 500$ for image datasets and $n_{\text{held}} = 100$ for tabular datasets. In addition, the optimization for the totally corrective step in line 7 of Alg.1 is addressed using an efficient proximal first-order solver for logistic loss with L1-regularization. In particular, the usage of different regularization strengths $\lambda_1, \lambda_2, \ldots, \lambda_k$ is handled by normalizing the prediction of base rule $\hbar_j$ for each training instance x_i as $\hbar_j(x_i)/\lambda_j$, for $j = 1, 2, \ldots, k$, and then using a constant strength for L1 regularization equal to 1.

Table 2: Dataset characteristics

	Dataset	Number of samples	Dimensions / Number of features
Image	Diabetic Retinopathy	2,076	224×224
	Fake and real face detection	2,041	224×224
	CIFAR-10 (cat vs dog)	12,000	32×32
	CIFAR-10 (car vs truck)	12,000	32×32
	MNIST (4 vs. 9)	11,791	28×28
	Food101 (pizza vs sushi)	2,000	512×384
	FashionMNIST (coat vs pullover)	14,000	28×28
	Breast histopathology	8,000	224×224
	Chest X-ray	5,218	224×224
Tabular	Ion	351	34
	Heart	303	13
	Diabetes	768	8
	German	1,000	24
	Credit	690	15
	Raisin	900	7
	Blood transfusion	748	4
	QSAR	1,055	41
	Haberman	306	3

For image datasets, inputs are resized to $224 \times 224 \times 3$ pixels and preprocessed following the standard ResNet-50 normalization protocol. All methods utilize the ResNet-50 backbone with pretrained weights that are fine-tuned using SGD-based training with learning rate 1×10^{-4}, a momentum of 0.9, and batch size of 32. The 'Ensemble' method (DeepEnsemble) (Lakshminarayanan et al., 2017) obtains different base rules by using a random initialization for the weights in the penultimate (dense) layer. For 'AdaBoost' and 'MBoost' methods each boosting round is initialized using the model weights from the previous round, as proposed in Wang et al. (2021).

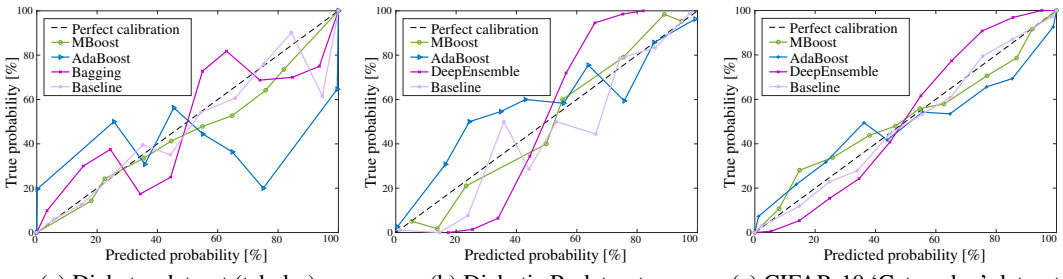
For tabular datasets, all methods use fully connected feedforward NNs with activation functions, layer sizes, and regularization determined using automated model selection based on Bayesian optimization (Snoek et al., 2012) through MATLAB's `fitcauto` function with time budget up to 3,600 seconds.

The results in Table 1 in the main paper are obtained using 10-fold stratified cross-validation. The ECE values measure calibration quality by partitioning predictions into 10 equal-width bins and computing the weighted average of the absolute difference between mean predicted probability and observed accuracy within each bin. The results in Fig. 1 in the main paper are obtained by averaging classification errors over 20 random data partitions. In each partition, $n = 2,500$ training samples are used to learn $K = 10$ base rules, which are fine-tuned for 10 epochs using the ResNet-50 backbone. In that figure, the actual generalization gaps $\boldsymbol{\lambda}^*$ of the base rules are determined using $8,500$ held-out samples distinct from the training samples. The cross-validated values of $\boldsymbol{\lambda}$ for the baseline comparison 'MBoost $\boldsymbol{\lambda}$ cross-val' are selected from the set $\{10^{-4}, 10^{-2}, 10^{0}, 10^{2}, 10^{4}\}$ using 5-fold partitions.

F.2 Additional experimental results

In the first set of additional experimental results, we further compare the performance of MBoost with benchmark methods.

The improvement in calibration is illustrated by Figure 3 that shows more detailed information than the ECE values. In particular, the figure shows the calibration curves obtained in several datasets corresponding to the results shown in Table 1. The results further show the improved calibration obtained by MBoost. In particular, AdaBoost method tends to be significantly overconfident, while Bagging and DeepEnsemble tend to be underconfident. Across different datasets, MBoost remains closer to the perfect calibration diagonal than the benchmark methods.



(a) Diabetes dataset (tabular) (b) Diabetic R. dataset (c) CIFAR-10 'Cat vs dog' dataset

(d) Fake and real face dataset (e) Chest X-ray dataset (f) MNIST dataset

Figure 3: Calibration curves for the methods compared on multiple datasets show that MBoost aligns more closely with the perfect calibration than state-of-the-art methods.

Table 3: Average classification error (left sub-table) and ECE (right sub-table) in $\%\pm$ standard deviation. Results compare Deep Boosting (DeepBoost), MBoost with decision-tree base rules (MBoost-Tree), and MBoost.

Dataset	DeepBoost	MBoost-Tree	MBoost	DeepBoost	MBoost-Tree	MBoost
Ion	8.0 ± 4.2	10.0 ± 5.2	8.57 ± 4.5	8.4 ± 3.9	9.7 ± 4.7	7.83 ± 2.8
Heart	23.1 ± 7.9	19.5 ± 6.5	13.7 ± 6.3	19.6 ± 6.5	17.0 ± 5.2	15.4 ± 3.6
Diabetes	26.4 ± 4.0	23.8 ± 6.4	21.8 ± 5.4	21.7 ± 3.3	13.9 ± 5.2	12.6 ± 3.2
German N.	26.6 ± 2.5	26.7 ± 2.7	23.9 ± 4.6	24.1 ± 2.4	14.2 ± 3.9	9.37 ± 2.3
Credit	15.8 ± 5.9	16.5 ± 7.1	12.5 ± 2.9	14.8 ± 5.9	11.9 ± 4.3	9.91 ± 2.0
Raisin	14.6 ± 4.0	14.9 ± 4.3	12.6 ± 2.8	13.4 ± 3.7	10.2 ± 2.6	9.48 ± 1.9
Blood T.	23.5 ± 1.8	22.6 ± 3.1	22.4 ± 3.9	22.1 ± 2.1	8.5 ± 3.4	12.2 ± 2.0
QSAR	17.9 ± 3.0	15.4 ± 3.0	10.3 ± 2.3	16.1 ± 2.9	9.1 ± 2.5	7.28 ± 1.2
Haberman	25.5 ± 3.8	30.1 ± 5.0	26.7 ± 5.2	23.0 ± 4.4	19.1 ± 4.4	15.1 ± 5.6

The results in the main paper compare the performance of methods that utilize NNs, since this is the type of strong rule most commonly used in practice. Table 3 complements those results by providing additional comparisons with Deep Boosting (Cortes et al., 2014), a method designed for decision trees that shares some methodological similarities with the proposed methods, as described in Section 2.1. For Deep Boosting, we use a Python reimplementation based on the authors' original code, available at `https://github.com/google/deepboost`, with the default settings provided in the public repository. For MBoost, we consider two implementations: one using the same setup as in Table 1 (MBoost), and another using the same family of base rules (decision trees) as Deep Boosting (MBoost-Tree). The results in Table 3 show that MBoost using NNs achieves substantially lower classification error than Deep Boosting, while MBoost-Tree achieves comparable performance when using the same family of base rules. The improvement achieved by MBoost is even more pronounced in terms of calibration, as the ECE is clearly lower for both MBoost implementations.

Table 4 shows the mean running time and standard deviation per dataset in minutes in comparison with the baseline and the state-of-the-art techniques that learn ensembles of base rules using 10 boosting rounds with 10 epochs per round. The table shows that MBoost does not require an increased complexity over state-of-the-art methods. In particular, the results in Table 4 show that the totally corrective updates do not introduce a significant computational burden.

Table 4: Average running time in minutes ± standard deviation for MBoost and state-of-the-art methods using multiple image (upper sub-table) and tabular (lower sub-table) datasets.

Dataset	Baseline 100 epochs	AdaBoost 10 rounds and 10 epochs	Ensemble 10 rounds and 10 epochs	MBoost 10 rounds and 10 epochs
Diabetic R.	3.03±.17	3.22±.05	8.93±.09	2.78±.14
F. and R. Face	3.22±.08	3.50±.11	9.42±.09	2.90±.90
Car vs truck	11.0±.12	11.9±.15	27.2±.11	12.4±.13
Cat vs dog	11.0±.11	12.4±.11	30.1±.12	12.4±.25
MNIST	10.5±.14	11.3±.15	29.4±.12	11.6±.18
Food101	3.58±.06	2.72±.07	12.3±.48	3.46±.11
F. MNIST	12.7±.15	13.8±.10	34.1±2.1	14.2±.31
Breast H.	7.94±.46	8.48±.08	21.9±.09	8.64±.06
Chest	6.53±.04	7.21±.21	17.2±.11	6.89±.09
Avg Time	1	1.07	2.74	1.08
Ion	1.64±.62	16.1±4.3	14.5±1.6	12.1±2.4
Heart	1.20±.43	17.2±2.6	21.1±3.1	12.9±3.6
Diabetes	2.25±1.1	51.7±3.1	40.8±3.2	42.5±4.6
German N.	2.34±.77	44.0±7.9	48.9±6.7	45.6±4.4
Credit	1.81±.82	47.8±.70	39.2±6.0	33.5±5.2
Raisin	3.74±1.5	59.4±3.3	54.8±6.4	54.2±4.2
Blood T.	3.73±2.3	57.5±.87	58.3±5.9	53.0±7.6
QSAR	4.14±2.0	59.7±7.4	50.7±3.2	54.6±3.3
Haberman	1.24±.30	35.1±4.1	45.1±6.7	28.7±7.7
Avg Time	1	17.6	16.9	15.3

F.3 Additional experimental results using synthetic data

In the second set of additional experimental results, we further evaluate the performance of MBoost using synthetic data generated as follows.

Let $x_{i,j}$ denote the value of the jth feature for the ith instance. Each feature is independent of the others and distributed following a standard normal distribution, that is, $x_{i,j} \sim \mathcal{N}(0,1)$ for $j = 1, 2, \ldots, d$ and we take $d = 100$. The binary labels $y \in \{-1, 1\}$ are determined by a non-linear interaction between the first ten features as

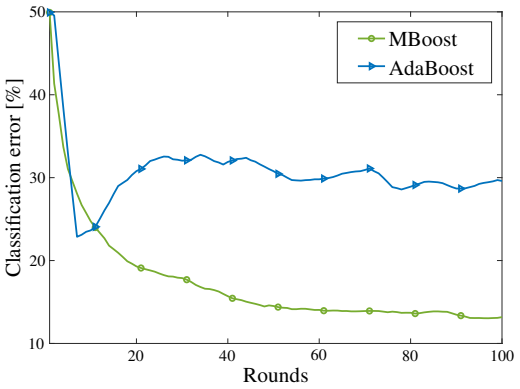
$$y = \text{sign}\Big(\Big(\sum_{j=1}^{5} x_j\Big)\Big(\sum_{j=6}^{10} x_j\Big)\Big). \tag{53}$$

This construction defines a challenging classification task where the decision boundary depends on second-order interactions of the first 10 features, while the remaining 90 features act as noise. This type of data is known as an index-2 model and is commonly used to illustrate the benefits of NNs with data that has hierarchical structure.

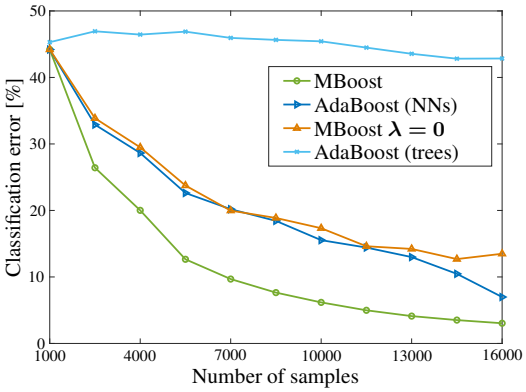
Figure 4a compares the evolution of classification error using MBoost and AdaBoost with NNs, showing that the presented approach can avoid overfitting where the existing method generalizes poorly. The results in Figure 4a are obtained by averaging the results over 10 repetitions. Both methods use $4,000$ training samples and fully connected feedforward NNs learned by using automated model selection.

Figure 4b shows the classification error obtained by varying the number of training samples. This figure illustrates how the proposed methods can quickly approach the Bayes risk as the number of samples increases. Furthermore, the figure shows the limitations of traditional boosting approaches using decision trees (with up to 10 splits). While MBoost approaches the Bayes risk asymptotically, AdaBoost with decision trees remains stagnant with an error rate above 40%. In the synthetic dataset considered, decision trees cannot capture the complex non-linear feature interactions present, whereas MBoost method with NNs successfully identifies and leverages the underlying structure.

Figure 5 further describes the benefits of the presented methods by showing the calibration curves obtained on the synthetic dataset for different training set sizes. Across all experimental settings (2,500, 4,000, and 10,000 training samples), MBoost exhibits the closest alignment to the ideal calibration, indicating more calibrated predictions than AdaBoost, Bagging, and the Baseline methods.

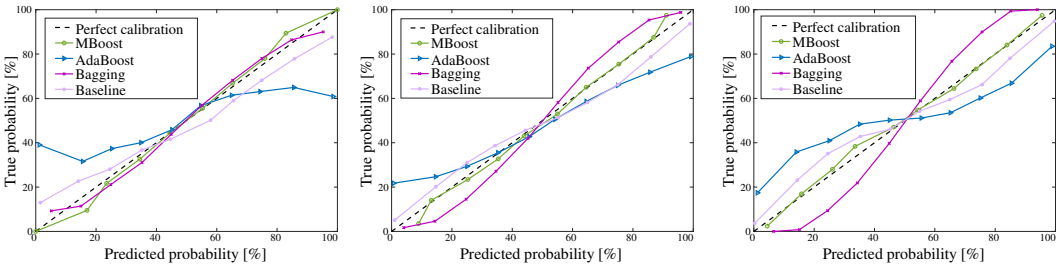


(a) MBoost error keeps decreasing over boosting rounds while AdaBoost overfits.



(b) MBoost with strong rules can converge to the Bayes risk in cases where boosting methods with decision trees are inadequate.

Figure 4: Results on synthetic data show the superior performance of the proposed MBoost method.



(a) $2,500$ training samples (b) $4,000$ training samples (c) $10,000$ training samples

Figure 5: Calibration curves on the synthetic dataset across increasing training set sizes show that MBoost aligns more closely with the perfect calibration diagonal than AdaBoost, Bagging, and the Baseline.

As the number of training samples increases, the calibration curve of MBoost approaches the diagonal more closely, suggesting that the method effectively leverages additional data while preserving its calibration quality. On the other hand, AdaBoost and Bagging tend to obtain overconfident and underconfident predictions, respectively, in accordance with the calibration results shown above for real-world datasets.

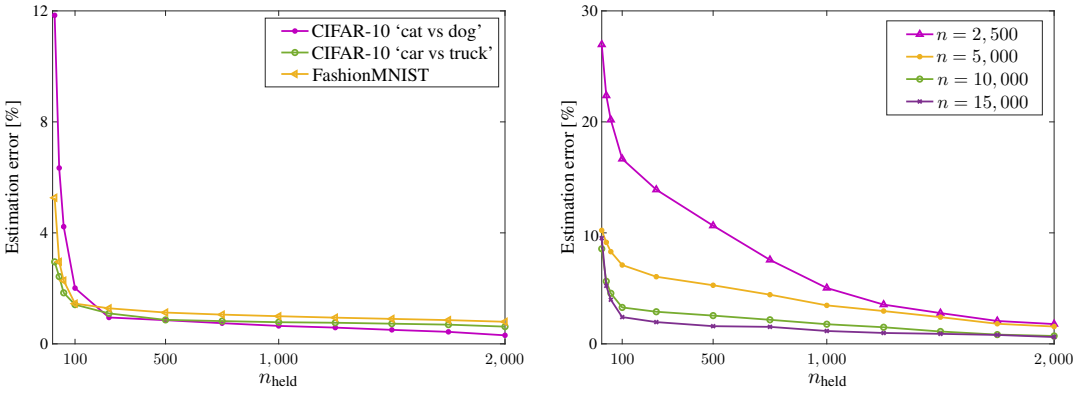
F.4 Reliability of estimates for the generalization gaps

In the third set of additional results, we assess the accuracy of the proposed data-driven estimates for generalization gaps and the impact of estimates on the final performance.

Figure 6 shows the accuracy obtained by the data-driven estimates varying the number of samples, n_{held}, used for the held-out estimates (line 6 of Alg. 1 in Section 5). The accuracy of the data-driven estimates is assessed as the difference between the generalization gap $\boldsymbol{\lambda}$ obtained as described in Section 5 and the actual generalization gap $\boldsymbol{\lambda}^*$ determined using 5,000 held-out samples distinct from the training samples. Fig. 6a shows the accuracy obtained using CIFAR-10 'cat vs dog', CIFAR-10 'car vs truck', and 'FashionMNIST' datasets with 5,000 training samples. Fig. 6b shows the accuracy obtained in the synthetic dataset for different training sizes.

The results in Figure 6 show the effectiveness of the approach proposed to estimate generalization gaps. In particular, the estimation errors significantly decrease with n_{held}. Held-out sizes on the order of a few hundred are sufficient to obtain accurate estimates, as corroborated also by the classification performance obtained in the experimental results shown in the rest of the paper.

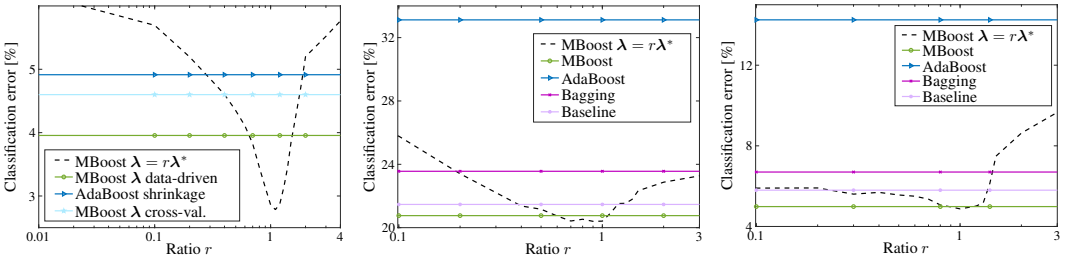
The impact of the generalization gap estimates on the final performance is shown in Figure 1 in the main paper using CIFAR-10 'cat vs dog' dataset and further assessed below by means of experimental results with synthetic data and the CIFAR-10 'car vs truck' dataset. Figure 7a is obtained by averaging classification errors over 20 random data partitions, similarly to Figure 1 in the main paper. Figures 7b and 7c are obtained using 360 seconds of automated model selection per boosting



(a) Real-world datasets (b) Synthetic dataset

Figure 6: Results on both real-world and synthetic data show that the error of estimates for the generalization gaps decreases rapidly with the number of held-out samples (n_{held}).

round, consistent with the settings described in Section 6. These figures show a similar behavior to that in Figure 1 in the main paper. In particular, MBoost provides strong performance as long as the values used for $\boldsymbol{\lambda}$ are not far from the actual generalization gaps. The figures also show that the presented data-driven estimates for generalization gaps can result in near-optimal performance close to that achieved using the actual generalization gaps.



(a) CIFAR-10 Car vs Truck (b) $2,500$ training samples (c) $10,000$ training samples

Figure 7: Results on real-world and synthetic data illustrate the impact of estimates for generalization gaps on classification performance. The figure also shows the performance that would be achieved using estimates proportional to the actual generalization gaps (black dashed curve). The use of constant values for generalization gaps results in poor performance but the use of data-driven estimates for individual generalization gaps can effectively leverage strong base rules.