Of Dice and Games: A Theory of Generalized Boosting

Marco Bressan MARCO.BRESSAN@UNIMI.IT
Università degli Studi di Milano, Italy

Nataly Brukhim NBRUKHIM@PRINCETON.EDU
Institute for Advanced Study, USA

Nicolò Cesa-Bianchi NICOLO.CESA-BIANCHI@UNIMI.IT
Università degli Studi di Milano, Italy
Politecnico di Milano, Italy

Emmanuel Esposito EMMANUEL@EMMANUELESPOSITO.IT
Università degli Studi di Milano, Italy

Yishay Mansour MANSOUR.YISHAY@GMAIL.COM
Tel Aviv University, Israel
Google Research

Shay Moran SMORAN@TECHNION.AC.IL
Technion, Israel
Google Research

Maximilian Thiessen MAXIMILIAN.THIESSEN@TUWIEN.AC.AT
TU Wien, Austria

Editors: Nika Haghtalab and Ankur Moitra

Abstract

Cost-sensitive loss functions are crucial in many real-world prediction problems, where different types of errors are penalized differently; for example, in medical diagnosis, a false negative prediction can lead to worse consequences than a false positive prediction. However, traditional PAC learning theory has mostly focused on the symmetric 0-1 loss, leaving cost-sensitive losses largely unaddressed. In this work we extend the celebrated theory of boosting to incorporate both cost-sensitive and multi-objective losses. Cost-sensitive losses assign costs to the entries of a confusion matrix, and are used to control the sum of prediction errors accounting for the cost of each error type. Multi-objective losses, on the other hand, simultaneously track multiple cost-sensitive losses, and are useful when the goal is to satisfy several criteria at once (e.g., minimizing false positives while keeping false negatives below a critical threshold).

We develop a comprehensive theory of cost-sensitive and multi-objective boosting, providing a taxonomy of weak learning guarantees that distinguishes which guarantees are trivial (i.e., can always be achieved), which ones are boostable (i.e., imply strong learning), and which ones are intermediate, implying non-trivial yet not arbitrarily accurate learning. For binary classification, we establish a dichotomy: a weak learning guarantee is either trivial or boostable. In the multiclass setting, we describe a more intricate landscape of intermediate weak learning guarantees. Our characterization relies on a geometric interpretation of boosting, revealing a surprising equivalence between cost-sensitive and multi-objective losses.

Keywords: Boosting, Minimax theorem, cost-sensitive learning, multi-objective learning, Blackwell's approachability

1. Introduction

In many machine learning applications, different types of mistakes may have very different consequences, making it crucial to consider the costs associated with them. For example, in medical diagnostics, failing to detect a serious illness (a false negative) can have life-threatening implications, whereas incorrectly diagnosing a healthy person as ill (a false positive) mostly leads to unnecessary stress and medical expenses. This disparity in error costs is not limited to binary decisions. For example, when recommending movies to a viewer with preferences "romance over action over horror", misclassifying a romance film as "horror" is probably worse than misclassifying it as "action". Besides weighting different kinds of mispredictions, one may even want to treat different kinds of mispredictions separately. That is, instead of a cost-sensitive criterion, one may use a multi-objective criterion, specifying acceptable rates for different types of mispredictions. For example, one may find acceptable a false positive rate of (say) 10% only if simultaneously the false negative rate is at most (say) 1%.

Despite the importance of misclassification costs in applications, the theoretical understanding of this setting is lacking. A glaring example, which motivates this work, is boosting. Broadly speaking, a boosting algorithm is a procedure that aggregates several *weak learners* (whose accuracy is only marginally better than a random guess) into a single *strong learner* (whose accuracy can be made as high as desired). Although this is a fundamental and well-studied machine learning technique, a theory of boosting accounting for cost-sensitive or multi-objective losses is missing, even in the simplest setting of binary classification.[1] In fact, if one can assign different costs to different kinds of mistakes, then even the meaning of "marginally better than a random guess" is not immediately clear; let alone the question of whether one can boost a weak learner to a strong learner, or what precisely "boosting" means. The present work addresses those challenges, providing a generalized theory of boosting which unifies different types of weak learning guarantees, including cost-sensitive and multi-objective ones, and extends the standard algorithmic boosting framework beyond the current state of the art. The fundamental question that we pose is:

Which cost-sensitive and/or multi-objective learners can be boosted? And how?

hen In classical boosting theory for binary classification, a sharp transition occurs at a weak learner's error threshold of $1/2$: if the weak learner is guaranteed to output hypotheses with an error rate below $1/2$, then it can be boosted to a strong learner with arbitrarily small error, for instance by AdaBoost (Freund and Schapire, 1997). Thus, a weak learning guarantee below $1/2$ implies strong learning. On the other hand, a guarantee of $1/2$ is trivial, as it can be achieved by tossing a fair coin—see Schapire and Freund (2012). Therefore, guarantees above $1/2$ are not boostable.

We investigate whether similar transitions exist for arbitrary cost-sensitive losses. A cost-sensitive loss w specifies the penalty $w(i, j)$ for predicting i when the true label is j, and can penalize prediction errors unequally (e.g., in binary classification, it may penalize false negatives more than false positives). Suppose we have access to a weak learner that outputs hypotheses with a cost-sensitive loss of at most $z > 0$ under w. For which values of z does this imply strong learning, so that the weak learner can be boosted to achieve arbitrarily small cost-sensitive loss according to

1. There are many works on adapting AdaBoost to cost-sensitive learning, but they do not address the fundamental question of identifying the minimal assumption on the cost-sensitive function which guarantees boosting. See more in Appendix D.

w? Which values of z are trivial, meaning they can always be achieved? Are there intermediate values of z that do not imply strong learning but are still non-trivial?

A similar question arises for multi-objective losses. A multi-objective loss is given by a vector $\boldsymbol{w} = (w_1, w_2, \ldots, w_r)$ where each w_i is a cost-sensitive loss as described above. For instance, in binary classification, a natural choice would be $\boldsymbol{w} = (w_n, w_p)$, where w_n measures false negatives and w_p false positives. Suppose again we have access to a weak learner that outputs hypotheses with loss at most $z_i \geq 0$ under w_i simultaneously for every i, forming a vector of guarantees $\boldsymbol{z} = (z_1, \ldots, z_r)$. Which $\boldsymbol{z}$ are trivial, in that they can always be achieved? Which $\boldsymbol{z}$ are boostable, allowing for a strong learner that achieves arbitrarily small error simultaneously for all of the losses w_i? And are there intermediate vectors $\boldsymbol{z}$ that fall between trivial and boostable?

We address these questions by introducing a new perspective on random guessing, framed as either a zero-sum game or a vector-payoff game (known as a Blackwell approachability game). This game-theoretic approach applies to both cost-sensitive and multi-objective learning, leading to a complete characterization of boostability in these cases. We then extend these techniques to the multiclass setting, where the boostability landscape becomes significantly more complex. While this perspective complements existing views of boosting as a zero-sum game, prior methods are not suited to the settings we examine here. The new tools introduced in this work effectively handle a broad learning framework, establishing a unified and comprehensive theory of generalized boosting.

In particular, we provide extensive answers to the above questions, as follows:

- *Cost-sensitive losses.* For binary classification, we establish a crisp dichotomy: each guarantee $z \geq 0$ is either trivial or boostable (Theorem A). We show that this transition occurs at a critical threshold given by the value of a zero-sum game defined by w itself. In the multiclass setting, the dichotomy expands into a hierarchy of guarantees, ranging from fully boostable to trivial. Here, we show that there exist multiple thresholds $0 < v_1(w) < v_2(w) < \cdots < v_\tau(w)$, where τ depends on the cost function w, and each guarantee $z \in (v_i, v_{i+1})$ can be boosted down to v_i (Theorem C). This generalizes the binary case, in which $\tau = 1$.

- *Multi-objective losses.* For binary classification, we again show a clean dichotomy; however, the threshold now takes a higher-dimensional form, becoming a surface that separates trivial guarantees from boostable ones (Theorem B). Figure 1 illustrates this surface for a loss vector $\boldsymbol{w}$ representing false positive and false negative errors. In the multiclass case, thresholds are determined by the boundaries of dice-attainable regions. Overall, we provide a taxonomy of learning guarantees, distinguishing those that are trivial (always achievable), boostable (implying strong learning), and intermediate (implying non-trivial yet not strong learning).

- *An equivalence between cost-sensitive and multi-objective losses.* We establish and exploit an equivalence between multi-objective and cost-sensitive losses that may be of independent interest (Theorem 7). Given a loss vector $\boldsymbol{w} = (w_1, \ldots, w_r)$, consider a weak learner that outputs hypotheses with loss at most z_i for each w_i. By linearity of expectation, it follows that for any convex combination $w_{\boldsymbol{\alpha}} = \sum \alpha_i w_i$, the weak learner's loss with respect to $w_{\boldsymbol{\alpha}}$ does not exceed $\sum \alpha_i z_i$. We prove that the converse also holds: if for each such $w_{\boldsymbol{\alpha}}$ there exists a weak learner with loss at most $\sum \alpha_i z_i$, then we can efficiently aggregate these learners into one that achieves loss at most z_i simultaneously for every w_i. Interestingly, a geometric perspective of this result reveals a connection to Blackwell's Approachability Theorem (Blackwell, 1956) and to scalarization methods in multi-objective optimization.

Organization of the manuscript. Section 2 gives a detailed walkthrough of our main results, their significance, and the underlying intuition. Section 3 gives the boosting algorithm for the binary cost-sensitive loss case. Appendix A addresses the binary classification case, in both the cost-sensitive and multi-objective flavors. Appendix B presents our equivalence connecting cost-sensitive learners with multi-objective learners. Finally, Appendix C considers the multiclass case.

2. Main Results

This section outlines the problem setting and the key questions we investigate, and provides an overview of our main results. We begin from the basic technical setup. We consider the standard PAC (Probably Approximately Correct) setting (Valiant, 1984), which models learning a concept class $\mathcal{F} \subseteq \mathcal{Y}^{\mathcal{X}}$ for a domain $\mathcal{X}$ and a label space $\mathcal{Y} = [k]$ with $k \geq 2$. We define a *cost function*, or simply *cost*, to be any function $w : \mathcal{Y}^2 \to [0,1]$ that satisfies $w(i,i) = 0$ for all $i \in \mathcal{Y}$;[2] the value $w(i,j)$ should be thought of as the penalty incurred by predicting i when the true label is j. Note that w can be seen as a $k \times k$ matrix indexed by $\mathcal{Y}$. For instance, for $\mathcal{Y} = \{-1,+1\}$, a valid cost is $w = \left(\begin{smallmatrix} 0 & 1 \\ 0.4 & 0 \end{smallmatrix}\right)$; this means that the cost of a false positive is $w(+1,-1) = 0.4$, and that of a false negative is $w(-1,+1) = 1$. For conciseness, in the binary setting we let $w_- = w(-1,+1)$ and $w_+ = w(+1,-1)$; and by some overloading of notation we denote by w both the matrix $\left(\begin{smallmatrix} 0 & w_- \\ w_+ & 0 \end{smallmatrix}\right)$ and the vector (w_+, w_-). For a cost w, a target function $f \in \mathcal{F}$, and a distribution $\mathcal{D}$ over $\mathcal{X}$, the cost-sensitive loss of a hypothesis $h : \mathcal{X} \to \mathcal{Y}$ is:

$$L_{\mathcal{D}}^{w}(h) \triangleq \mathbb{E}_{x \sim \mathcal{D}}\Big[w(h(x), f(x))\Big]. \tag{1}$$

For example, when $w(i,j) = \mathbb{I}\{i \neq j\}$, then $L_{\mathcal{D}}^{w}(h)$ simply corresponds to $\mathbb{P}_{x \sim \mathcal{D}}(h(x) \neq f(x))$, the standard 0-1 loss used in binary classification. When h is a randomized hypothesis, and thus for a given x the prediction $h(x)$ is a random variable, the definition of $L_{\mathcal{D}}^{w}(h)$ takes the expectation over the randomization of h as well.

We begin by presenting our contributions for the binary case (Section 2.1), followed by their extension to the multiclass case (Section 2.2).

2.1. Binary setting

Let $\mathcal{X}$ be any domain, let $\mathcal{Y} = \{-1,+1\}$, and let $\Delta_{\mathcal{Y}}$ be the set of all distributions over $\mathcal{Y}$. We begin by defining a suitable notion of "weak learner" with respect to a cost function w.

Definition 1 ((w,z)-learner) *Let $w : \mathcal{Y}^2 \to [0,1]$ be a cost function and let $z \geq 0$. An algorithm $\mathcal{A}$ is a (w,z)-learner for $\mathcal{F} \subseteq \mathcal{Y}^{\mathcal{X}}$ if there is a function $m_0 : (0,1)^2 \to \mathbb{N}$ such that for every $f \in \mathcal{F}$, every distribution $\mathcal{D}$ over $\mathcal{X}$, and every $\epsilon, \delta \in (0,1)$ the following claim holds. If S is a sample of $m_0(\epsilon,\delta)$ examples drawn i.i.d. from $\mathcal{D}$ and labeled by f, then $\mathcal{A}(S)$ returns a predictor $h : \mathcal{X} \to \Delta_{\mathcal{Y}}$ such that $L_{\mathcal{D}}^{w}(h) \leq z + \epsilon$ with probability at least $1 - \delta$.*

In the above definition of (w,z)-learner we admit randomized predictors as output, which we model as predictors that map each point in $\mathcal{X}$ to a distribution over the label space $\mathcal{Y}$. Note that one can always use such an h to predict a single label on input x by independently sampling a label from $h(x) \in \Delta_{\mathcal{Y}}$ at each call; see Appendix F for more discussion.

2. Although the range of the cost is $[0,1]$ our results can be easily extended to arbitrary costs $w : \mathcal{Y}^2 \to \mathbb{R}_{\geq 0}$.

	binary	**multiclass**
0-1 loss	$\frac{1}{2}$	$\frac{1}{2}, \frac{2}{3}, \ldots, \frac{k-1}{k}$
cost w	$V(w) = \frac{w_+ w_-}{w_+ + w_-}$	$v_2(w), \ldots, v_\tau(w)$
multi-objective $\boldsymbol{w}$	coin-attainable boundary of $C(\boldsymbol{w})$	J-dice-attainable boundaries of $D_J(\boldsymbol{w})$

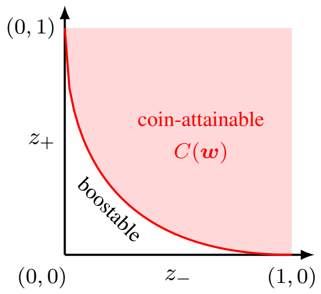

Figure 1: Boostability thresholds. *Binary case.* Under the standard 0-1 loss, it is well-known that the boostability threshold is $\frac{1}{2}$: any value below it can be boosted, while any value above it is trivially attainable by non-boostable learners. Theorem A generalizes this dichotomy by showing that the boostability threshold under an arbitrary cost function w is $\mathrm{V}(w)$, see Equation (3). For multi-objective losses, the boostability threshold takes instead the form of the boundary of the coin-attainable region $C(\boldsymbol{w})$; see Definition 5 and Theorem B. The plot on the right shows $C(\boldsymbol{w})$ when $\boldsymbol{w}$ consists of two objectives, the false-positive rate and the false-negative rate. *Multiclass case.* Under the standard 0-1 loss, boostability is known to be determined by the thresholds $\frac{1}{2}, \frac{2}{3}, \ldots, \frac{k-1}{k}$ (Brukhim et al., 2023a). Theorem C shows that, more in general, for a cost function w the boostability thresholds are given by the scalars $v_n(w)$ defined in Equation (6). For multi-objective losses, the boostability thresholds take the form of boundaries of dice-attainable regions $D_J(\boldsymbol{w})$, see Definition 10 and Theorem D.

We study the question of whether a (w, z)-learner can be boosted. Broadly speaking, a learner $\mathcal{A}$ is *boostable* if there exists a learning algorithm $\mathcal{B}$ that can achieve arbitrarily small loss by aggregating a small set of predictors obtained from $\mathcal{A}$. Typically, this works as follows: given black-box access to $\mathcal{A}$ and a labeled sample S, algorithm $\mathcal{B}$ invokes $\mathcal{A}$ on subsamples $S_1, \ldots, S_T$ of S to produce weak predictors $h_1, \ldots, h_T$. It then outputs a predictor $\bar{h}(x) = g(h_1(x), \ldots, h_T(x))$ using some aggregation function $g : \Delta_{\mathcal{Y}}^T \to \Delta_{\mathcal{Y}}$. For instance, in classic binary boosting the function g is usually a weighted majority vote. In this work, we consider arbitrary such aggregations. The goal is to ensure that the loss of the aggregate predictor, $L_{\mathcal{D}}^w(\bar{h})$, goes to zero with T, resulting in a $(w, 0)$-learner. Our guiding question can then be stated as:

Can we boost a (w, z)-learner to a $(w, 0)$-learner?

In order to develop some intuition, let us consider again the case when w yields the standard 0-1 loss. In that case, Definition 1 boils down to the standard notion of a weak PAC learner.[3] Then, classic boosting theory states that every (w, z)-learner with $z < 1/2$ can be boosted to a $(w, 0)$-learner; and it is easy to see that a loss of $1/2$ is always achievable, simply by predicting according to a fair coin toss. Thus, the value $1/2$ yields a sharp dichotomy: every $z < 1/2$ can be boosted and drawn to 0, while every $z \geq 1/2$ is trivially achievable and cannot be brought below $1/2$.

Can this classic dichotomy between "boostable" and "trivial" be extended to accommodate arbitrary cost functions? It turns out that this can be done if one uses as trivial predictors *biased* coins.

3. Notice that this definition is slightly more general than the classic weak PAC learning definition, in which $z = 1/2 - \gamma$ and $\epsilon = 0$, yet we consider learners which are allowed to be arbitrarily close to z.

Indeed, we show that, by taking such random guessing strategies into account, one can identify a general boostability threshold for all cost functions w. However, in contrast to the 0-1 loss case, this critical threshold between the boostable and non-boostable guarantees is no longer $1/2$; instead, it is a function of the cost w. More precisely, the threshold is determined by the outcome of a simple two-player zero-sum game, as we describe next.

The game involves a minimizing player (the predictor) and a maximizing player (the environment). The minimizing player selects a distribution $\boldsymbol{p}$ over $\mathcal{Y} = \{-1, +1\}$; similarly, the maximizing player selects a distribution $\boldsymbol{q}$ over $\mathcal{Y} = \{-1, +1\}$. The payoff matrix of the game is w. The overall cost paid by the predictor is then:

$$w(\boldsymbol{p}, \boldsymbol{q}) \triangleq \mathbb{E}_{i\sim \boldsymbol{p}}\mathbb{E}_{j\sim \boldsymbol{q}}\, w(i, j)\,. \tag{2}$$

Following standard game theory, the *value of the game* is the scalar:

$$\mathrm{V}(w) \triangleq \min_{\boldsymbol{p}\in\Delta_{\mathcal{Y}}}\ \max_{\boldsymbol{q}\in\Delta_{\mathcal{Y}}} w(\boldsymbol{p}, \boldsymbol{q})\,. \tag{3}$$

In words, $\mathrm{V}(w)$ is the smallest loss that the predictor can ensure by tossing a biased coin (i.e., $\boldsymbol{p}$) without knowing anything about the true distribution of the labels (i.e., $\boldsymbol{q}$). Now consider a value $z \geq 0$. It is easy to see that, if $z \geq \mathrm{V}(w)$, then there exists a universal (w, z)-learner—one that works for all $\mathcal{F}$ and all distributions $\mathcal{D}$ over $\mathcal{X}$: this is the learner that, ignoring the input sample, returns the predictor $h_{\boldsymbol{p}}$ such that $h_{\boldsymbol{p}}(x) = \boldsymbol{p}$ for all $x \in \mathcal{X}$. Indeed, by Equation (3) the loss of $h_{\boldsymbol{p}}$ is at most $\mathrm{V}(w)$ regardless of $f \in \mathcal{F}$ and of the distribution $\mathcal{D}$. Formally, we have:

Definition 2 (Random guess) *A hypothesis h is a* random guess *if its prediction is independent of the input point $x \in \mathcal{X}$. Equivalently, h is a random guess if there exists a probability distribution $\boldsymbol{p} \in \Delta_{\mathcal{Y}}$ such that $h(x) = \boldsymbol{p}$ for every $x \in \mathcal{X}$.*

We prove that the non-boostability condition $z \geq \mathrm{V}(w)$ is tight. That is, we prove that every (w, z)-learner with $z < \mathrm{V}(w)$ *is* boostable to a $(w, 0)$-learner. Thus, the value of the game $\mathrm{V}(w)$ given by Equation (3) is precisely the threshold for boostability. Formally:

Theorem A (Cost-sensitive boosting, binary case) *Let $\mathcal{Y} = \{-1, +1\}$ and let $w = (w_+, w_-) \in (0, 1]^2$ be a cost. Then, for all $z \geq 0$, exactly one of the following holds.*

- *(w, z) **is boostable**: for every $\mathcal{F} \subseteq \mathcal{Y}^{\mathcal{X}}$, every (w, z)-learner is boostable to a $(w, 0)$-learner.*
- *(w, z) **is trivial**: there exists a random guess h such that the learner that always outputs h is a (w, z)-learner for every $\mathcal{F} \subseteq \mathcal{Y}^{\mathcal{X}}$.*

Moreover, (w, z) is boostable if and only if $z < V(w)$, where $\mathrm{V}(w) = \frac{w_+ w_-}{w_+ + w_-}$.

The proof of Theorem A is given in Appendix A.1. In a nutshell, Theorem A says that anything that beats the weakest possible learner (a coin) is as powerful as a the strongest possible learner; between these two extremes there is no middle ground. Remarkably, the proof of Theorem A is simple and yet it relies on *three* distinct applications of von Neumann's Minimax Theorem (von Neumann, 1928). The first application, which also appears in classical binary boosting, is used to aggregate a distribution over the weak learners. The other two applications are unique to the cost-sensitive setting: one arises in the analysis of the boosting algorithm (first item of Theorem A), and

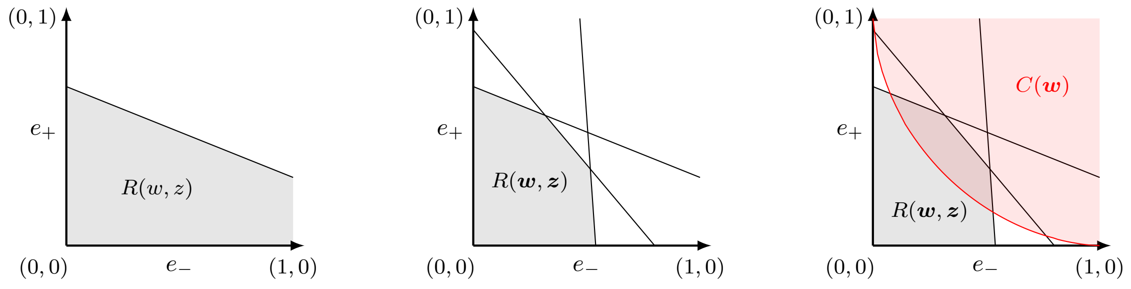

Figure 2: The feasible region (Definition 4) and its relationship with the coin-attainable region (Definition 5) for the binary case, as given by Theorem B. Each point in the plane is a pair $\boldsymbol{e} = (e_+, e_-)$ where e_+ and e_- represent respectively a false-positive error rate and false-negative error rate. The first plot shows the feasible region $R(w, z)$ for a single cost-sensitive loss w and a scalar guarantee $z \in [0, 1]$; this is the set of all points $\boldsymbol{e}$ such that $\langle \boldsymbol{e}, w\rangle \leq z$. The middle plot shows the feasible region $R(\boldsymbol{w}, \boldsymbol{z})$ for a multi-objective cost $\boldsymbol{w} = (w_1, w_2, w_3)$ and a vector $\boldsymbol{z} \in [0, 1]^3$; this is the set $R(w_1, z_1) \cap R(w_2, z_2) \cap R(w_3, z_3)$. The right plot shows in addition the coin-attainable region $C(\boldsymbol{w})$, i.e., the set of all points $\boldsymbol{e}$ attainable by a trivial learner. By Theorem B, in the binary case $C(\boldsymbol{w})$ is precisely the set of all $\boldsymbol{e}$ satisfying $\sqrt{e_+} + \sqrt{e_-} \geq 1$, and if $C(\boldsymbol{w}) \cap R(\boldsymbol{w}, \boldsymbol{z}) \neq \emptyset$, as is the case in the plot, then there exist $(\boldsymbol{w}, \boldsymbol{z})$-learners that cannot be boosted.

the other one in defining the trivial learner (second item of Theorem A). These last two applications are both based on the zero-sum game defined above.

We also note that the first item of Theorem A is obtained constructively by an efficient boosting algorithm (Algorithm 1), as we detail in Appendix A. Unlike standard boosting approaches, the final prediction on any input point x is not the majority vote over the h_t's, the weak learning hypotheses. Instead, the prediction is constructed by computing the mass assigned to -1 and $+1$ by the h_t's, weighted by, respectively, w_- and w_+. The smallest weighted mass wins, and the corresponding label is deterministically returned. We show that this ensures correct deterministic labeling of the entire sample with the desired probability.

2.1.1. MULTI-OBJECTIVE LOSSES

A multi-objective loss is specified by a multi-objective cost, that is, a vector $\boldsymbol{w} = (w_1, \dots, w_r)$ where $w_i : \mathcal{Y}^2 \to [0, 1]^2$ is a cost for every $i = 1, \dots, r$. This allows one to express several criteria by which a learner is evaluated; for example, it allows us to measure separately the false positive rate and the false negative rate, giving a more fine-grained control over the error bounds. The guarantees of a learner can then be expressed by bounding $L_{\mathcal{D}}^{w_i}(h)$ by some $z_i \geq 0$ simultaneously for each $i = 1, \dots, r$. This leads to the following generalization of Definition 1.

Definition 3 ($(\boldsymbol{w}, \boldsymbol{z})$-learner) *Let $r \in \mathbb{N}$, let $\boldsymbol{w} = (w_1, \dots, w_r)$ where each $w_i : \mathcal{Y}^2 \to [0, 1]^2$ is a cost function, and let $\boldsymbol{z} \in [0, 1]^r$. An algorithm $\mathcal{A}$ is a $(\boldsymbol{w}, \boldsymbol{z})$-learner for $\mathcal{F} \subseteq \mathcal{Y}^{\mathcal{X}}$ if there is a function $m_0 : (0, 1)^2 \to \mathbb{N}$ such that for every $f \in \mathcal{F}$, every distribution $\mathcal{D}$ over $\mathcal{X}$, and every $\epsilon, \delta \in (0, 1)$ what follows holds. If S is a sample of $m_0(\epsilon, \delta)$ examples drawn i.i.d. from $\mathcal{D}$ and labeled by f, then $\mathcal{A}(S)$ returns a predictor $h : \mathcal{X} \to \Delta_{\mathcal{Y}}$ such that with probability at least $1 - \delta$:*

$$\forall\, i = 1, \dots, r, \qquad L_{\mathcal{D}}^{w_i}(h) \leq z_i + \epsilon\ .$$

Consider for example $\boldsymbol{w} = (w_p, w_n)$ with $w_p = \left(\begin{smallmatrix} 0 & 0 \\ 1 & 0 \end{smallmatrix}\right)$ and $w_n = \left(\begin{smallmatrix} 0 & 1 \\ 0 & 0 \end{smallmatrix}\right)$. Then w_p counts the false positives, w_n the false negatives. Thus a $(\boldsymbol{w}, \boldsymbol{z})$-learner for, say, $\boldsymbol{z} = (0.1, 0.4)$ ensures simultaneously a false-positive rate arbitrarily close to 0.1 and false-negative arbitrarily close to 0.4. See Figure 2 for illustrative geometric examples.

As shown in Figure 2, any general multi-objective guarantee $(\boldsymbol{w}, \boldsymbol{z})$ can be viewed as a set of linear constraints, determined by (w_i, z_i). This set identifies a feasible region in $[0,1]^2$, formally defined next.

Definition 4 ($(\boldsymbol{w}, \boldsymbol{z})$-feasible region) *Let $\boldsymbol{w} = (w_1, \ldots, w_r)$, and let $\boldsymbol{z} \in [0,1]^r$. Then, the feasible region $R(\boldsymbol{w}, \boldsymbol{z})$ is defined by:*

$$R(\boldsymbol{w}, \boldsymbol{z}) = \{\boldsymbol{e} \in [0,1]^2 : \forall i = 1, \ldots, r, \;\; \langle \boldsymbol{e}, w_i \rangle \le z_i\}.$$

Like for the cost-sensitive case, we address the question of which $(\boldsymbol{w}, \boldsymbol{z})$-learners are boostable. In other words we ask: given $\boldsymbol{w}$, what is the right "threshold" for $\boldsymbol{z}$? Equivalently, for which $\boldsymbol{z}$ can we always boost a $(\boldsymbol{w}, \boldsymbol{z})$-learner to a $(\boldsymbol{w}, \boldsymbol{0})$-learner, and for which $\boldsymbol{z}$ do there exist $(\boldsymbol{w}, \boldsymbol{z})$-learners that are not boostable? It turns out that the answer is more nuanced than in the cost-sensitive case of Section 2.1, and yet we can prove the same "all or nothing" phenomenon of Theorem A, as depicted in Figure 1. In particular, every $\boldsymbol{z} \in [0,1]^r$ is either entirely boostable, in the sense that every $(\boldsymbol{w}, \boldsymbol{z})$-learner can be boosted to a $(\boldsymbol{w}, \boldsymbol{0})$-learner, or *trivial*, in the sense that there exists a $(\boldsymbol{w}, \boldsymbol{z})$-learner whose output is always a hypothesis that can be simulated by a (biased) random coin. To this end we introduce the notion of *coin attainability*. This is the multi-objective equivalent of the condition $z \ge \mathrm{V}(w)$ for w being a scalar cost as in Section 2.1.

Definition 5 (Coin attainability) *Let $\boldsymbol{w} = (w_1, \ldots, w_r)$ where $w_i : \mathcal{Y}^2 \to [0,1]$ is a cost function for every $i = 1, \ldots, r$, and let $\boldsymbol{z} \in [0,1]^r$. We say that $(\boldsymbol{w}, \boldsymbol{z})$ is* coin-attainable *if*

$$\forall \boldsymbol{q} \in \Delta_{\mathcal{Y}}, \; \exists \boldsymbol{p} \in \Delta_{\mathcal{Y}}, \; \forall i = 1, \ldots, r, \qquad w_i(\boldsymbol{p}, \boldsymbol{q}) \le z_i \; . \tag{4}$$

The coin-attainable region *of $\boldsymbol{w}$, denoted by $C(\boldsymbol{w})$, is the set of all $\boldsymbol{z}$ s.t. $(\boldsymbol{w}, \boldsymbol{z})$ is coin-attainable.*

It is not hard to see that, if $\boldsymbol{z}$ is coin-attainable, then there exists a universal trivial $(\boldsymbol{w}, \boldsymbol{z})$-learner for all $\mathcal{F} \subseteq \mathcal{Y}^{\mathcal{X}}$. Fix indeed any distribution $\mathcal{D}$ over $\mathcal{X}$ and any $f \in \mathcal{F}$. First, one can learn the marginal $\boldsymbol{q}$ of $\mathcal{D}$ over $\mathcal{Y}$ within, say, total variation distance ϵ; then, by the coin-attainability of $\boldsymbol{z}$, one can return a hypothesis h such that $h(x) = \boldsymbol{p}$, where $\boldsymbol{p}$ ensures $w_i(\boldsymbol{p}, \boldsymbol{q}) \le z_i + \epsilon$ for all $i = 1, \ldots, r$. Formally, recalling the definition of random guess (Definition 2), we have:

Definition 6 (Trivial learner) *A learning algorithm $\mathcal{A}$ is* trivial *if it only outputs random guesses.*

Thus, a trivial learner can (at best) learn the *best* random guess and return it. Clearly, such a learner is in general not boostable to a $(\boldsymbol{w}, \boldsymbol{0})$-learner. Then, we prove the following.

Theorem B (Multi-objective boosting, binary case) *Let $\mathcal{Y} = \{-1, 1\}$, let $\boldsymbol{w} = (w_1, \ldots, w_r)$ where $w_i : \mathcal{Y}^2 \to [0,1]$ is a cost, and let $\boldsymbol{z} \in [0,1]^r$. Then, exactly one of the following holds:*

- $(\boldsymbol{w}, \boldsymbol{z})$ ***is boostable****: for every $\mathcal{F} \subseteq \mathcal{Y}^{\mathcal{X}}$, every $(\boldsymbol{w}, \boldsymbol{z})$-learner is boostable to a $(\boldsymbol{w}, \boldsymbol{0})$-learner.*
- $(\boldsymbol{w}, \boldsymbol{z})$ ***is trivial****: there exists a trivial learner that is a $(\boldsymbol{w}, \boldsymbol{z})$-learner for all $\mathcal{F} \subseteq \mathcal{Y}^{\mathcal{X}}$.*

Moreover, $(\boldsymbol{w}, \boldsymbol{z})$ is boostable if and only if $\boldsymbol{z} \notin C(\boldsymbol{w})$. Equivalently, $(\boldsymbol{w}, \boldsymbol{z})$ is boostable if and only if every $(a, b) \in R(\boldsymbol{w}, \boldsymbol{z})$ satisfies $\sqrt{a} + \sqrt{b} < 1$.

The proof of Theorem B is given in Appendix A.4, and is based on a certain scalarization of the loss amenable to a reduction to Theorem A. Let us spend a few words on Theorem B. First, note how Definition 5 is reminiscent of the zero-sum game of Section 2.1. The crucial difference, however, is that now the maximizing player (the environment) plays first, and the minimizing player (the predictor) can then choose $\boldsymbol{p}$ as a function of $\boldsymbol{q}$. As shown in more detail in Appendix A.2, this is essential for proving the second item—that is, for coin-attainability to capture exactly the guarantees attainable by trivial learners. For the scalar-valued game of Section 2.1, instead, by von Neumann's Minimax Theorem the order of the players is immaterial to the value of the game $\mathrm{V}(w)$.

Second, note that the multi-objective setting is more general than that of a (scalar-valued) zero-sum game, as here the payoff is vector-valued. This can be thought of as a Blackwell approachability game (Blackwell, 1956), a generalization of zero-sum games to vectorial payoffs. Indeed, it turns out that there is a connection between these two notions, in the sense of Blackwell's theorem: we prove that a certain vectorial bound $\boldsymbol{z}$ can be attained (with respect to a vector-valued loss $\boldsymbol{w}$) if an only if all the "scalarizations" of $\boldsymbol{z}$ can be attained with respect to the corresponding scalarizations of $\boldsymbol{w}$. In fact, we can show that this is captured formally by an equivalence between cost-sensitive learners and multi-objective learners, as explained in Section 2.1.2.

Finally, the relationship given by Theorem B between the feasible region $R(\boldsymbol{w}, \boldsymbol{z})$, the coin-attainable region $C(\boldsymbol{w})$, and the boostability of $(\boldsymbol{w}, \boldsymbol{z})$-learners can be given a simple and intuitive geometric interpretation, as described in Figure 2.

2.1.2. COST-SENSITIVE VS. MULTI-OBJECTIVE LOSSES: A GENERAL EQUIVALENCE

We establish a general, formal connection between cost-sensitive and multi-objective learning. The starting point is the straightforward observation that, by definition, a $(\boldsymbol{w}, \boldsymbol{z})$-learner is a (w_i, z_i)-learner for every $i = 1, \dots, r$. Does the converse hold, too? That is, if for each $i = 1, \dots, r$ there exist a (w_i, z_i)-learner $\mathcal{A}_i$, can we conclude that there exists a $(\boldsymbol{w}, \boldsymbol{z})$-learner $\mathcal{A}$? Perhaps surprisingly, we can show that this holds if we consider all *convex combinations* of the scalar guarantees (w_i, z_i). Formally, given a distribution $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_r) \in \Delta_r$, let $w_{\boldsymbol{\alpha}} = \sum_{i=1}^r \alpha_i w_i$ and $z_{\boldsymbol{\alpha}} = \langle \boldsymbol{\alpha}, \boldsymbol{z} \rangle$. It is easy to see that the observation above continues to hold: a $(\boldsymbol{w}, \boldsymbol{z})$-learner is a $(w_{\boldsymbol{\alpha}}, z_{\boldsymbol{\alpha}})$-learner for every such $\boldsymbol{\alpha}$. The next result shows that the converse holds, too: the existence of a $(w_{\boldsymbol{\alpha}}, z_{\boldsymbol{\alpha}})$-learner for every $\boldsymbol{\alpha} \in \Delta_r$ implies the existence of a $(\boldsymbol{w}, \boldsymbol{z})$-learner.

Theorem 7 (Equivalence of learners) *Let $\mathcal{F} \subseteq \mathcal{Y}^{\mathcal{X}}$, let $\boldsymbol{w} = (w_1, \dots, w_r)$, and let $\boldsymbol{z} \in [0, 1]^r$. Then, the following are equivalent.*

1. *There exists a $(\boldsymbol{w}, \boldsymbol{z})$-learner for $\mathcal{F}$.*
2. *For every $\boldsymbol{\alpha} \in \Delta_r$ there exists a $\big(w_{\boldsymbol{\alpha}}, z_{\boldsymbol{\alpha}}\big)$-learner for $\mathcal{F}$.*

Moreover, the equivalence is obtained constructively by an efficient algorithm.

We remark that Theorem 7 holds in the multiclass case, too (i.e., for arbitrary sets $\mathcal{Y}$). Thus, the interplay between multi-objective and cost-sensitive learning is a general phenomenon, not limited to binary classifiers. The reduction from multi-objective to cost-sensitive in Theorem 7 resembles the weighted sum scalarization method for multi-objective optimization (Ehrgott, 2005, Chapter

3). Although scalarization is a popular technique in multi-objective learning (Jin and Sendhoff, 2008; Lin et al., 2019; Hu et al., 2024), it is unclear whether these results can be used to provide further insights for our problem. In the same vein as Theorem 7, we prove an equivalence between trivial guarantees of cost-sensitive learners (see Theorem A) and trivial guarantees of multi-objective learners (see Theorem B).

Theorem 8 (Equivalence of trivial guarantees) *Let $\boldsymbol{w} = (w_1, \dots, w_r)$ and let $\boldsymbol{z} \in [0,1]^r$. Then the following are equivalent.*

1. *$(\boldsymbol{w}, \boldsymbol{z})$ is coin-attainable.*
2. *For every $\boldsymbol{\alpha} \in \Delta_r$ it holds that $z_{\boldsymbol{\alpha}} \geq \mathrm{V}(w_{\boldsymbol{\alpha}})$.*

As Theorem 7, we note that Theorem 8 also holds in the general multiclass case. The proof of Theorem 8 again uses von Neumann's Minimax Theorem, based on a two-player game that differs from those described earlier. In this game, the pure strategies of the maximizing player are the different elements of $[r]$, representing the various objectives of the cost function. The proofs of all equivalence theorems are given in Section B.

2.2. Multiclass setting

For binary classification tasks, we get an especially clean and refined mathematical picture, in which boostability is fully determined by a single threshold, i.e., $\mathrm{V}(w)$. That is, any learner with loss value below it can be boosted, while any value above it is attainable by a trivial non-boostable learner. In the unweighted case, recent work by Brukhim et al. (2023a) demonstrated that a *multichotomy* emerges, governed by $k-1$ thresholds: $\frac{1}{2}, \frac{2}{3}, \dots, \frac{k-1}{k}$, where k is the number of labels (see Theorem 3, Brukhim et al. (2023a)). For each such threshold, any learner achieving a loss below it can be "partially" boosted to the next threshold below it, but not further than that. Here, we extend their results to arbitrary cost functions and prove that a similar phenomenon holds more generally.

In contrast to the unweighted case, the landscape of boostability in the general setting is more complex, consisting of many thresholds corresponding to distinct possible subsets of labels. Interestingly, the thresholds of boostability can be computed precisely, and are all determined by the outcome of zero-sum games, as defined next. We generalize the zero-sum game defined in Equation (2) to the case of $k \geq 2$ labels. It turns out that in the multiclass setting, a more refined zero-sum game is needed, introducing a certain asymmetry between the sets of allowable strategies for each player. In particular, we restrict the set of distributions $\boldsymbol{q}$ which the maximizing player is allowed to choose. For a subset $J \subseteq \mathcal{Y}$ of labels define the value of the game *restricted to the subset* J as:

$$\mathrm{V}_J(w) \triangleq \min_{\boldsymbol{p} \in \Delta_{\mathcal{Y}}} \max_{\boldsymbol{q} \in \Delta_J} w(\boldsymbol{p}, \boldsymbol{q}) \ . \tag{5}$$

Importantly, only the maximizing player is restricted to Δ_J while the minimizing player is not. Thus, $\mathrm{V}_J(w)$ is the smallest loss one can ensure by predicting with a die *given that the correct label is in J*. Clearly, for every $J' \subseteq J$ it holds that $\mathrm{V}_{J'}(w) \leq \mathrm{V}_J(w)$. Consider the subsets $J \subseteq \mathcal{Y}$ in nondecreasing order of $\mathrm{V}_J(w)$.[4] We then denote each such cost by $v_s(w)$ for some $s \in \{1, \dots, 2^k\}$, so that overall we have,

$$0 = v_1(w) < v_2(w) < \dots < v_\tau(w) = \mathrm{V}_{\mathcal{Y}}(w) \ , \tag{6}$$

4. For notational easiness we define $\mathrm{V}_\emptyset(w) = 0$.

where $\tau \leq 2^k$ depends on the cost function w. The first equality holds since $\mathrm{V}_{\{y\}}(w) = 0$ for $y \in \mathcal{Y}$. Moreover, by a straightforward calculation it can be shown that, for the unweighted case in which w is simply the 0-1 loss (i.e., $w(i,j) = \mathbb{I}\{i \neq j\}$), the thresholds $\mathrm{V}_J(w)$ specialize to those given by Brukhim et al. (2023a). This is described in the following fact.

Fact 9 *For the 0-1 loss w, every $n \geq 1$, and $J \in \binom{\mathcal{Y}}{n}$, it holds that $\mathrm{V}_J(w) = 1 - \frac{1}{n}$.*

Thus, for the 0-1 loss we have $\tau = k$. Note that although for the general case there is no closed-form expression for the value of each threshold, it can each be determined by solving the corresponding linear program representation of the zero-sum game described above.

We can now state our main results on multiclass cost-sensitive boosting, given next.

Theorem C (Cost-sensitive boosting, multiclass case) *Let $\mathcal{Y} = [k]$, let $w : \mathcal{Y}^2 \to [0,1]$ be any cost function, and let $z \geq 0$. Let $v_1 < v_2 < \cdots < v_\tau$ as defined in Equation (6),[5] and let n be the largest integer such that $v_n \leq z$. Then, the following claims both hold:*

- (w,z) ***is n-boostable:*** *for every $\mathcal{F} \subseteq \mathcal{Y}^\mathcal{X}$, every (w,z)-learner is boostable to (w, v_n)-learner.*
- (w,z) ***is not $(n-1)$-boostable:*** *there exist a class $\mathcal{F} \subseteq \mathcal{Y}^\mathcal{X}$ and a (w,z)-learner for $\mathcal{F}$ that cannot be boosted to (w,z')-learner for $\mathcal{F}$ for any $z' < v_n$.*

The proof of Theorem C is given in Appendix C. In words, Theorem C implies that there is a partition of the interval $[0,1]$, on which the loss values lie, into at most τ sub-intervals or "buckets", based on v_n. Then, any loss value z in a certain bucket $\big[v_n, v_{n+1}\big)$ can be boosted to the lowest value within the bucket, but not below that.

The above boostability result is obtained constructively via an efficient algorithm, as described in Appendix C. At a high-level, the algorithm works in two phases. In the first phase, it turns a (w,z)-learner into a list learner, based on the framework of *List PAC learning* (Brukhim et al., 2022; Charikar and Pabbaraju, 2023). That is, for each threshold v_n, any learner achieving a loss below it can be "boosted" to a predictor that rules out *some* incorrect labels, albeit not all in general. Thus, for every example x the predictor returns a subset of candidate labels, also called a *list*, where its size depends on v_n. Then, in the second phase, the algorithm turns a list-learner into a (w, v_n)-learner by sampling from the list returned by the list-learner.

2.2.1. MULTICLASS CLASSIFICATION, MULTI-OBJECTIVE LOSS

We shift our focus to *multi-objective* losses in the multiclass setting. Recall the notion of $(\boldsymbol{w}, \boldsymbol{z})$-learner from Definition 3. Our goal is to prove a multi-objective counterpart of Theorem C; that is, to understand for which $\boldsymbol{z}'$ a $(\boldsymbol{w}, \boldsymbol{z})$-learner can be boosted to a $(\boldsymbol{w}, \boldsymbol{z}')$-learner. Unlike the scalar cost setting of Theorem C, however, the set of those $\boldsymbol{z}'$ that are reachable by boosting a $(\boldsymbol{w}, \boldsymbol{z})$-learner is not a one-dimensional interval in $[0,1]$, but a subset of $[0,1]^r$.

As a first step, we generalize the notion of coin attainability given by Definition 5. The key ingredient that we shall add is to restrict the distribution of the true labels over a given set $J \subseteq \mathcal{Y}$. From the point of view of a random guesser, this amounts to knowing that the correct label must be in J. We then say that a guarantee is J-dice-attainable if a trivial learner can satisfy that guarantee over any distribution supported only over J.

5. When clear from context we omit w and denote a threshold by v_n.

Definition 10 (J-dice-attainability) *Let $\mathcal{Y} = [k]$, $\boldsymbol{w} = (w_1, \dots, w_r)$ where each $w_i \colon \mathcal{Y}^2 \to [0,1]$ is a cost function, $\boldsymbol{z} \in [0,1]^r$, and let $J \subseteq \mathcal{Y}$ be nonempty. Then $(\boldsymbol{w}, \boldsymbol{z})$ is J-*dice-attainable *if:*

$$\forall \boldsymbol{q} \in \Delta_J, \ \exists \boldsymbol{p} \in \Delta_{\mathcal{Y}}, \ \forall i = 1, \dots, r, \qquad w_i(\boldsymbol{p}, \boldsymbol{q}) \le z_i \ . \tag{7}$$

*The J-*dice-attainable region *of $\boldsymbol{w}$ is the set $D_J(\boldsymbol{w}) \triangleq \{\boldsymbol{z} : (\boldsymbol{w}, \boldsymbol{z}) \text{ is } J\text{-dice-attainable}\}$.*

Intuitively, if $(\boldsymbol{w}, \boldsymbol{z})$ is J-dice-attainable, then a $(\boldsymbol{w}, \boldsymbol{z})$-learner is not better than a random die over J, and therefore should not be able to separate any label in J. Using this notion, we define a partial ordering over $[0,1]^r$. For every $\boldsymbol{z}, \boldsymbol{z}' \in [0,1]^r$ we write $\boldsymbol{z} \preceq_{\boldsymbol{w}} \boldsymbol{z}'$ when

$$\forall J \subseteq \mathcal{Y}, \ \ \boldsymbol{z} \in D_J(\boldsymbol{w}) \implies \boldsymbol{z}' \in D_J(\boldsymbol{w}) \, , \tag{8}$$

and $\boldsymbol{z} \not\preceq_{\boldsymbol{w}} \boldsymbol{z}'$ otherwise. Intuitively, $\boldsymbol{z} \preceq_{\boldsymbol{w}} \boldsymbol{z}'$ means that, whenever $\boldsymbol{z}$ is not better than a die, then $\boldsymbol{z}'$ is not better than a die either. We prove that this partial ordering precisely characterizes the boostability of $\boldsymbol{z}$ to $\boldsymbol{z}'$. We can now state our main result for multi-objective multiclass boosting.

Theorem D (Multi-objective boosting, multiclass case) *Let $\mathcal{Y} = [k]$, let $\boldsymbol{w} = (w_1, \dots, w_r)$ where each $w_i \colon \mathcal{Y}^2 \to [0,1]$ is a cost function, and let $\boldsymbol{z} \in [0,1]^r$. Then, the following claims both hold:*

- $(\boldsymbol{w}, \boldsymbol{z})$ ***is boostable to*** $(\boldsymbol{w}, \boldsymbol{z}')$ ***for every*** $\boldsymbol{z} \preceq_{\boldsymbol{w}} \boldsymbol{z}'$*: for every class $\mathcal{F} \subseteq \mathcal{Y}^{\mathcal{X}}$, every $(\boldsymbol{w}, \boldsymbol{z})$-learner for $\mathcal{F}$ is boostable to a $(\boldsymbol{w}, \boldsymbol{z}')$-learner for $\mathcal{F}$.*
- $(\boldsymbol{w}, \boldsymbol{z})$ ***is not boostable to*** $(\boldsymbol{w}, \boldsymbol{z}')$ ***for any*** $\boldsymbol{z} \not\preceq_{\boldsymbol{w}} \boldsymbol{z}'$*: there exist a class $\mathcal{F} \subseteq \mathcal{Y}^{\mathcal{X}}$ and a $(\boldsymbol{w}, \boldsymbol{z})$-learner for $\mathcal{F}$ that is a trivial learner and, therefore, cannot be boosted to a $(\boldsymbol{w}, \boldsymbol{z}')$-learner for $\mathcal{F}$ for any $\boldsymbol{z}'$ such that $\boldsymbol{z} \not\preceq_{\boldsymbol{w}} \boldsymbol{z}'$.*

Let us elaborate briefly on Theorem D. In the multiclass cost-sensitive setting, where the overall cost is a scalar, the landscape of boostability is determined by a sequence of (totally ordered) scalar thresholds $v_n \in [0,1]$; and those thresholds determine whether a (w, z)-learner can be boosted to a (w, z')-learner, for $z' < z$, as detailed in Theorem C. In the multiclass multi-objective setting, those scalar thresholds are replaced by a set of surfaces in $[0,1]^r$. Each such surface corresponds to the boundary of the dice-attainable regions $D_J(\boldsymbol{w})$. Those surfaces can be seen as thresholds of a higher-dimensional form, separating between different boostability guarantees.

Proof sketch The boosting algorithm which attains the first item of Theorem D is given in Appendix C. A high-level proof sketch is described next. Let $\boldsymbol{\alpha} \in \Delta_r$, and denote $w_{\boldsymbol{\alpha}} = \boldsymbol{\alpha} \cdot \boldsymbol{w}$ and $z'_{\boldsymbol{\alpha}} = \boldsymbol{\alpha} \cdot \boldsymbol{z}'$. Observe that if, for every $\boldsymbol{\alpha} \in \Delta_r$, we can prove that a $(\boldsymbol{w}, \boldsymbol{z})$-learner $\mathcal{A}$ can be used to obtain a $(w_{\boldsymbol{\alpha}}, z'_{\boldsymbol{\alpha}})$-learner for $\mathcal{F}$, then by Theorem 7 this implies the existence of a $(\boldsymbol{w}, \boldsymbol{z}')$-learner as well, which completes the proof. Thus, it remains to prove that for every $\boldsymbol{\alpha} \in \Delta_r$, our $(\boldsymbol{w}, \boldsymbol{z})$-learner $\mathcal{A}$ can be converted to a $(w_{\boldsymbol{\alpha}}, z'_{\boldsymbol{\alpha}})$-learner.

First, the algorithm will convert the learner $\mathcal{A}$ into a list learner (as in the first step of the proof of Theorem C), but such that the list learner always returns lists in the set $\mathrm{av}(\boldsymbol{w}, \boldsymbol{z}) \triangleq \{J \subseteq \mathcal{Y} : \boldsymbol{z} \notin D_J(\boldsymbol{w})\}$. We call this learner a $\mathrm{av}(\boldsymbol{w}, \boldsymbol{z})$-*list learner*, as it never outputs the "bad" sets which can be avoided, i.e., sets for which the pair $(\boldsymbol{w}, \boldsymbol{z})$ is not dice-attainable.

Second, to obtain the $\mathrm{av}(\boldsymbol{w}, \boldsymbol{z})$-list learner, the algorithm first obtains multiple standard list learners, where each one is responsible for a single "bad" list J, for which $\boldsymbol{z} \notin D_J(\boldsymbol{w})$. We can obtain such a list learner by techniques used for the proof of Theorem C. Once we have such a list learner for each "bad" J, we take the *intersection of all lists*. We then prove that the resulting algorithm is indeed a $\mathrm{av}(\boldsymbol{w}, \boldsymbol{z})$-list learner.

Lastly, it is not hard to show that for any $\boldsymbol{\alpha} \in \Delta_r$, the $\mathrm{av}(\boldsymbol{w}, \boldsymbol{z})$-list learner we just constructed is also a $(w_{\boldsymbol{\alpha}}, z'_{\boldsymbol{\alpha}})$-learner, by appropriately sampling from the returned list. ■

3. Binary cost-sensitive boosting

This section describes at a high level the boosting algorithm behind Theorem A; see Appendix A for the full proofs. For convenience, as $\mathcal{Y} = \{-1, +1\}$, we will encode the cost function $w : \mathcal{Y}^2 \to [0, 1]$ as a 2-by-2 matrix $\left(\begin{smallmatrix} 0 & w_- \\ w_+ & 0 \end{smallmatrix}\right)$, where $w_+ = w(+1, -1)$ is the cost of a false positive and $w_- = w(-1, +1)$ the cost of a false negative. We will also use the fact, provable through easy calculations, that the value of the game $\mathrm{V}(w)$ equals 0 if $\min(w_-, w_+) = 0$ and $\frac{w_- w_+}{w_- + w_+}$ otherwise. Before moving forward, we need a measure of the advantage that a (w, z)-learner has compared to a coin.

Definition 11 (Margin, binary case) *The* margin *of a (w, z)-learner is $\gamma \triangleq \max\{0, \mathrm{V}(w) - z\}$.*

When $w_+ = w_- = 1$, and thus w is the standard 0-1 cost, γ is the usual notion of margin that defines a weak learner in binary classification tasks. In that case, it is well known that to boost a weak learner with margin γ it is necessary and sufficient to invoke the learner a number of times that scales with $1/\gamma^2$. This remains true for general w and z, in that our boosting algorithm below invokes the (w, z)-learner a number of times proportional to $1/\gamma^2$.

Algorithm 1 gives the boosting algorithm that turns any (w, z)-learner $\mathcal{A}$ with $z < \mathrm{V}(w)$ into a $(w, 0)$-learner $\mathcal{A}'$. The algorithm takes in input a labeled sample, the learner (w, z)-learner $\mathcal{A}$, and some additional parameters that determine its behaviour. The main result we prove is:

Theorem 12 *Let $\mathcal{Y} = \{-1, +1\}$ and $\mathcal{F} \subseteq \mathcal{Y}^{\mathcal{X}}$. If $\mathcal{A}$ is a (w, z)-learner for $\mathcal{F}$ with margin $\gamma > 0$, then Algorithm 1 with the choice of parameters of Lemma 13 is a $(w, 0)$-learner for $\mathcal{F}$. Under the same assumptions, Algorithm 1 makes $T = O\big(\frac{\ln(m)}{\gamma^2}\big)$ oracle calls to $\mathcal{A}$, where m is the size of the input sample, and has sample complexity $m(\epsilon, \delta) = \widetilde{O}\Big(m_0\big(\frac{\gamma}{3}, \frac{\delta}{2T}\big) \cdot \frac{\ln(1/\delta)}{\epsilon \cdot \gamma^2}\Big)$, where m_0 is the sample complexity of $\mathcal{A}$.*

It is immediate to see that Theorem 12 implies the first item of Theorem A. The proof of Theorem 12, given in Appendix A, consists of two steps: proving that Algorithm 1 returns a hypothesis consistent with the input sample (Lemma 13), and showing generalization by a compression argument (Lemma 14).

At a high level, Algorithm 1 follows standard boosting procedures. Given a (w, z)-learner $\mathcal{A}$ and a labeled sample S, Algorithm 1 uses $\mathcal{A}$ to obtain a sequence of hypotheses $h_1, \ldots, h_T$ whose average has loss close to z on each single example in S. This can be achieved through any regret-minimizing algorithm, but for the sake of concreteness our algorithm uses a modified version of Hedge (Freund and Schapire, 1997) where the update step re-weights examples as a function of w. Unlike standard approaches, the final prediction on a point x is not just the majority vote of the h_t's. Instead, the prediction is constructed by computing the mass of h_t's returning -1, weighted by w_-, and the mass of h_t's returning $+1$, weighted by w_+. These two weighted masses,

Algorithm 1 Boosting a binary (w, z)-learner

```
Input: sample $S = (x_i, y_i)_{i=1}^m$; $(w, z)$-learner $\mathcal{A}$; parameters $T, \eta, \widehat{m}$
1: Initialize: $D_1(i) = 1$ for all $i = 1, ..., m$.
2: for $t = 1, \ldots, T$ do
3:     Compute the distribution $\mathcal{D}_t \triangleq \frac{D_t}{\sum_i D_t(i)}$ over $[m]$.
4:     Draw a set $S_t$ of $\widehat{m}$ labeled examples i.i.d. from $\mathcal{D}_t$ and obtain $h_t = \mathcal{A}(S_t)$.
5:     For every $i = 1, \ldots, m$ let:
           $$D_{t+1}(i) \triangleq D_t(i) \cdot e^{\eta \cdot w(h_t(x_i), y_i)} .$$
6: end for
7: For all $x \in \mathcal{X}$ let $\boldsymbol{p}^x \triangleq \frac{1}{T} \sum_{t=1}^T h_t(x) \in \Delta_{\mathcal{Y}}$.
8: return $\widehat{h}_S : \mathcal{X} \to \mathcal{Y}$ such that, for all $x \in \mathcal{X}$,
           $$\widehat{h}_S(x) \triangleq \begin{cases} +1 & \text{if } w(\boldsymbol{p}^x, +1) < V(w) , \\ -1 & \text{otherwise.} \end{cases}$$
```

which in the algorithm appear as $w(\boldsymbol{p}^x, +1)$ and $w(\boldsymbol{p}^x, -1)$, are the expected w-loss over x of the predictor obtained by averaging the h_t's, when the true label of the point x is respectively $+1$ and -1. When w is the standard 0-1 loss, those masses are just the plain fraction of h_t's that on x predict respectively -1 and $+1$. The final predictor $\widehat{h}_S$ then returns the label y with smaller weighted mass, or, equivalently, such that $w(\boldsymbol{p}^x, y) < V(w)$. When w is the standard 0-1 loss, $V(w) = \frac{1}{2}$ and the predictor just returns the plain majority label, recovering the classic Hedge algorithm. It can be shown that the output predictor $\widehat{h}_S$ ensures correct labeling of the entire sample with the desired probability. Together with a generalization bound based on compression arguments, this yields Theorem 12.

Acknowledgments

MB, NCB, and EE are partially supported by the MUR PRIN grant 2022EKNE5K (Learning in Markets and Society), the FAIR (Future Artificial Intelligence Research) project, funded by the NextGenerationEU program within the PNRR-PE-AI scheme, the EU Horizon CL4-2022-HUMAN-02 research and innovation action under grant agreement 101120237, project ELIAS (European Lighthouse of AI for Sustainability), and the One Health Action Hub, University Task Force for the resilience of territorial ecosystems, funded by Università degli Studi di Milano (PSR 2021-GSA-Linea 6).

SM is supported by a Robert J. Shillman Fellowship, by ISF grant 1225/20, by BSF grant 2018385, by an Azrieli Faculty Fellowship, by Israel PBC-VATAT, and by the Technion Center for Machine Learning and Intelligent Systems (MLIS), and by the European Union (ERC, GENERALIZATION, 101039692). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them.

YM has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement number 882396), by the Israel Science Foundation, the Yandex Initiative for Machine Learning at Tel Aviv University and a grant from the Tel Aviv University Center for AI and Data Science (TAD).

MT acknowledges support from a DOC fellowship of the Austrian academy of sciences (ÖAW) and travel support from ELISE (GA number 951847).

References

Erin L Allwein, Robert E Schapire, and Yoram Singer. Reducing multiclass to binary: A unifying approach for margin classifiers. *Journal of Machine Learning Research*, 1:113–141, 2000.

Noga Alon, Steve Hanneke, Ron Holzman, and Shay Moran. A theory of PAC learnability of partial concept classes. In *Annual Symposium on Foundations of Computer Science (FOCS)*, 2022.

Shai Ben-David, Philip M Long, and Yishay Mansour. Agnostic boosting. In *International Conference on Computational Learning Theory*, pages 507–516. Springer, 2001.

D Blackwell. An analog of the minimax theorem for vector payoffs. *Pacific Journal of Mathematics*, 6(1):1–8, 1956.

Nataly Brukhim, Elad Hazan, Shay Moran, and Robert E. Schapire. Multiclass boosting and the cost of weak learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, NeurIPS, 2021.

Nataly Brukhim, Daniel Carmon, Irit Dinur, Shay Moran, and Amir Yehudayoff. A characterization of multiclass learnability. In *Annual Symposium on Foundations of Computer Science (FOCS)*, 2022.

Nataly Brukhim, Amit Daniely, Yishay Mansour, and Shay Moran. Multiclass boosting: simple and intuitive weak learning criteria. In *Advances in Neural Information Processing Systems (NeurIPS)*, NeurIPS, 2023a.

Nataly Brukhim, Steve Hanneke, and Shay Moran. Improper multiclass boosting. In *Conference on Learning Theory (COLT)*, 2023b.

Clément L. Canonne. A short note on learning discrete distributions, 2020.

Moses Charikar and Chirag Pabbaraju. A characterization of list learnability. In *Symposium on Theory of Computing (STOC)*, 2023.

Alexander Durgin and Brendan Juba. Hardness of improper one-sided learning of conjunctions for all uniformly falsifiable CSPs. In *Algorithmic Learning Theory (ALT)*, 2019.

Matthias Ehrgott. *Multicriteria optimization*. Springer Science & Business Media, 2005.

Charles Elkan. The foundations of cost-sensitive learning. In Bernhard Nebel, editor, *International Joint Conference on Artificial Intelligence (IJCAI)*, 2001.

Wei Fan, Salvatore J. Stolfo, Junxin Zhang, and Philip K. Chan. Adacost: Misclassification cost-sensitive boosting. In *International Conference on Machine Learning (ICML)*, 1999.

Vitaly Feldman. Distribution-specific agnostic boosting. *arXiv preprint arXiv:0909.2927*, 2009.

Yoav Freund. Boosting a weak learning algorithm by majority. In *Computational Learning Theory (COLT)*, 1990.

Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.*, 55(1):119–139, 1997.

Kasper Green Larsen and Martin Ritzert. Optimal weak to strong learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Nika Haghtalab, Michael Jordan, and Eric Zhao. A unifying perspective on multi-calibration: Game dynamics for multi-objective learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Yuzheng Hu, Ruicheng Xian, Qilong Wu, Qiuling Fan, Lang Yin, and Han Zhao. Revisiting scalarization in multi-task learning: A theoretical perspective. *Advances in Neural Information Processing Systems (NeurIPS)*, 36, 2024.

Jiyan Jiang, Wenpeng Zhang, Shiji Zhou, Lihong Gu, Xiaodong Zeng, and Wenwu Zhu. Multi-objective online learning. In *International Conference on Learning Representations (ICLR)*, 2023.

Yaochu Jin. *Multi-objective machine learning*. Springer Science & Business Media, 2007.

Yaochu Jin and Bernhard Sendhoff. Pareto-based multiobjective machine learning: An overview and case studies. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 38(3):397–415, 2008.

Adam Tauman Kalai and Rocco A. Servedio. Boosting in the presence of noise. *J. Comput. Syst. Sci.*, 71(3):266–290, 2005.

Adam Tauman Kalai, Yishay Mansour, and Elad Verbin. On agnostic boosting and parity learning. In *Symposium on Theory of Computing (STOC)*, 2008.

Adam Tauman Kalai, Varun Kanade, and Yishay Mansour. Reliable agnostic learning. *Journal of Computer and System Sciences*, 78(5):1481–1495, 2012.

Varun Kanade and Adam Kalai. Potential-based agnostic boosting. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2009.

Varun Kanade and Justin Thaler. Distribution-independent reliable learning. In *Conference on Learning Theory (COLT)*, 2014.

Grigoris I. Karakoulas and John Shawe-Taylor. Optimizing classifers for imbalanced training sets. In *Advances in Neural Information Processing Systems (NeurIPS)*, 1998.

M. Kearns. Thoughts on hypothesis boosting. Unpublished, December 1988.

Jyrki Kivinen. Learning reliably and with one-sided error. *Mathematical systems theory*, 28(2): 141–172, 1995.

Iago Landesa-Vazquez and José Luis Alba-Castro. Shedding light on the asymmetric learning capability of adaboost. *Pattern Recognition Letters*, 33(3):247–255, 2012.

Iago Landesa-Vazquez and José Luis Alba-Castro. Double-base asymmetric adaboost. *Neurocomputing*, 118:101–114, 2013.

Iago Landesa-Vázquez and José Luis Alba-Castro. Revisiting adaboost for cost-sensitive classification. part i: Theoretical perspective. *arXiv preprint arXiv:1507.04125*, 2015.

Xi Lin, Hui-Ling Zhen, Zhenhua Li, Qing-Fu Zhang, and Sam Kwong. Pareto multi-task learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 32, 2019.

Charles X Ling and Victor S Sheng. Cost-sensitive learning and the class imbalance problem. *Encyclopedia of Machine Learning*, 2011:231–235, 2008.

Charles X. Ling and Victor S. Sheng. Cost-sensitive learning. In *Encyclopedia of Machine Learning and Data Mining*. Springer, 2017.

Philip M. Long and Rocco A. Servedio. Adaptive martingale boosting. In Daphne Koller, Dale Schuurmans, Yoshua Bengio, and Léon Bottou, editors, *Advances in Neural Information Processing Systems (NeurIPS)*, 2008.

Shiyin Lu, Guanghui Wang, Yao Hu, and Lijun Zhang. Multi-objective generalized linear bandits. *arXiv preprint arXiv:1905.12879*, 2019.

Hamed Masnadi-Shirazi and Nuno Vasconcelos. Cost-sensitive boosting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(2):294–309, 2011.

Mikael Møller Høgsgaard, Kasper Green Larsen, and Markus Engelund Mathiasen. The many faces of optimal weak-to-strong learning. *arXiv e-prints*, pages arXiv–2408, 2024.

Indraneel Mukherjee and Robert E Schapire. A theory of multiclass boosting. *Journal of Machine Learning Research*, 14:437–497, 2011.

Nikolaos Nikolaou, Narayanan Unny Edakunni, Meelis Kull, Peter A. Flach, and Gavin Brown. Cost-sensitive boosting algorithms: Do we really need them? *Machine Learning*, 104(2-3): 359–384, 2016.

Vinod Raman and Ambuj Tewari. Online agnostic multiclass boosting. *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Vinod Raman, Daniel T Zhang, Young Hun Jung, and Ambuj Tewari. Online boosting for multilabel ranking with top-k feedback. *arXiv preprint arXiv:1910.10937*, 2019.

Sivan Sabato and Naftali Tishby. Multi-instance learning with any hypothesis class. *The Journal of Machine Learning Research*, 13(1):2999–3039, 2012.

Robert E. Schapire. The strength of weak learnability. *Machine Learning*, 5(2):197–227, 1990.

Robert E Schapire and Yoav Freund. *Boosting: Foundations and algorithms*. Cambridge university press, 2012.

Robert E Schapire and Yoram Singer. Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, 37(3):297–336, 1999.

Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge University Press, 2014.

Yanmin Sun, Mohamed S. Kamel, Andrew K. C. Wong, and Yang Wang. Cost-sensitive boosting for classification of imbalanced data. *Pattern Recognition*, 40(12):3358–3378, 2007.

Kai Ming Ting. A comparative study of cost-sensitive boosting algorithms. In *International Conference on Machine Learning (ICML)*, 2000.

Leslie G Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

Paul A. Viola and Michael J. Jones. Fast and robust classification using asymmetric adaboost and a detector cascade. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2001.

John von Neumann. Zur Theorie der Gesellschaftsspiele. *Mathematische Annalen*, 100(1):295–320, 1928.

Bianca Zadrozny, John Langford, and Naoki Abe. Cost-sensitive learning by cost-proportionate example weighting. In *International Conference on Data Mining (ICDM)*, 2003.

Appendix A. Binary Classification

This section considers the classic binary classification setting, where $\mathcal{Y} = \{-1, +1\}$. Appendix A.1 tackles the simplest case, where the cost w is scalar, and explores the conditions that make a (w, z)-learner boostable, proving Theorem A. Appendix A.2 considers multi-objective costs, where the cost $\boldsymbol{w}$ is vector-valued, for a specific but illustrative choice of $\boldsymbol{w}$, proving a special case of Theorem B and of Theorem 8. The reason for restricting $\boldsymbol{w}$ to a special form is that this allows us to introduce the key ideas while avoiding heavy notation. Finally, in Appendix A.4 we give the full proof of Theorem B. For convenience, as $\mathcal{Y} = \{-1, +1\}$, we will sometimes encode a cost function $w : \mathcal{Y}^2 \to [0, 1]$ as a 2-by-2 matrix $\left(\begin{smallmatrix} 0 & w_- \\ w_+ & 0 \end{smallmatrix}\right)$, where $w_+ = w(+1, -1)$ is the cost of a false positive and $w_- = w(-1, +1)$ the cost of a false negative. We will also use the fact, provable through easy calculations, that the value of the game $\mathrm{V}(w)$ equals 0 if $\min(w_-, w_+) = 0$ and $\frac{w_- w_+}{w_- + w_+}$ otherwise. When $w_+ = w_- = 1$, and thus w is the standard 0-1 cost, γ is the usual notion of margin that defines a weak learner in binary classification tasks. In that case, it is well known that to boost a weak learner with margin γ it is necessary and sufficient to invoke the learner a number of times that scales with $1/\gamma^2$. This remains true for general w and z, in that our boosting algorithm below invokes the (w, z)-learner a number of times proportional to $1/\gamma^2$.

A.1. Boosting a (w, z)-learner

This section proves Theorem 12. It is immediate to see that Theorem 12 implies the first item of Theorem A; for the second item see Lemma 15 below. In the rest of this subsection we prove Theorem 12. We do so in two steps: first we prove that Algorithm 1 returns a hypothesis consistent with the input sample (Lemma 13), and then we show generalization by a compression argument (Lemma 14).

Lemma 13 *Let $\mathcal{Y} = \{-1, +1\}$ and let $w : \mathcal{Y}^2 \to [0, 1]$ be a cost function. Let $\mathcal{F} \subseteq \mathcal{Y}^{\mathcal{X}}$, and let $\mathcal{A}$ be a (w, z)-learner for $\mathcal{F}$ with margin $\gamma > 0$ and sample complexity m_0. Fix any $f \in \mathcal{F}$, let $\mathcal{D}$ be any distribution over $\mathcal{X}$, and let $S = \{(x_1, y_1), \dots, (x_m, y_m)\}$ be a multiset of m examples given by i.i.d. points $x_1, \dots, x_m \sim \mathcal{D}$ labeled by f. Finally, fix any $\delta \in (0, 1)$. If Algorithm 1 is given S, oracle access to $\mathcal{A}$, and parameters $T = \left\lceil \frac{18 \ln(m)}{\gamma^2} \right\rceil$, $\eta = \sqrt{\frac{2\ln(m)}{T}}$, and $\widehat{m} = m_0\left(\frac{\gamma}{3}, \frac{\delta}{T}\right)$, then Algorithm 1 makes T calls to $\mathcal{A}$ and returns $\widehat{h}_S : \mathcal{X} \to \mathcal{Y}$ such that with probability at least $1 - \delta$:*

$$\widehat{h}_S(x_i) = y_i \qquad \forall i = 1, \dots, m\ .$$

Proof Fix any $i \in [m]$. Given that $\max_{t,i} w(h_t(x_i), y_i) \le 1$ and that $\eta \le 1$, the standard analysis of Hedge (Freund and Schapire, 1997) shows that:

$$\sum_{t=1}^{T} w(h_t(x_i), y_i) \le \frac{\ln(m)}{\eta} + \frac{\eta}{2} T + \sum_{t=1}^{T} \mathbb{E}_{j \sim \mathcal{D}_t}\Big[w(h_t(x_j), y_j)\Big]\ . \tag{9}$$

Dividing both sides by T, and using the definitions of η and T at the right-hand side, yields:

$$\frac{1}{T}\sum_{t=1}^{T} w(h_t(x_i), y_i) \le \frac{\gamma}{3} + \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}_{j \sim \mathcal{D}_t}\Big[w(h_t(x_j), y_j)\Big]\ . \tag{10}$$

For every $t = 1, \dots, T$, since $h_t = \mathcal{A}(S_t)$, since $\mathcal{A}$ is a (w, z)-learner for $\mathcal{F}$, and by the choice of $\widehat{m}$:

$$\mathbb{P}\left(\mathbb{E}_{j\sim\mathcal{D}_t}\Big[w(h_t(x_j), y_j)\Big] > z + \frac{\gamma}{3}\right) \le \frac{\delta}{T} \,. \tag{11}$$

By averaging over T and taking a union bound over $t = 1, \dots, T$, we have that, with probability at least $1 - \delta$,

$$\frac{1}{T}\sum_{t=1}^{T} \mathbb{E}_{j\sim\mathcal{D}_t}\Big[w(h_t(x_j), y_j)\Big] \le z + \frac{\gamma}{3} \,. \tag{12}$$

Now suppose Equation (12) holds. We shall prove that $\widehat{h}_S(x_i) = y_i$ for every $i \in [m]$. First, Equation (10) together with the definition of γ implies that for every $i \in [m]$:

$$w(\boldsymbol{p}^{x_i}, y_i) = \frac{1}{T}\sum_{t=1}^{T} w(h_t(x_i), y_i) \le \frac{2}{3}\gamma + z \le \mathrm{V}(w) - \frac{\gamma}{3} < \mathrm{V}(w) \,, \tag{13}$$

where the first equality holds by definition of $\boldsymbol{p}^x$ from Algorithm 1 and by linearity of w.

Now we claim that, for every $x \in \mathcal{X}$, we have $w(\boldsymbol{p}^x, y) < V(w)$ for at most one $y \in \{-1, +1\}$. To this end observe that, for any $x \in \mathcal{X}$,

$$w(\boldsymbol{p}^x, +1) = w(\boldsymbol{p}^x, +1) = w_- \cdot p^x(+1) \,, \tag{14}$$
$$w(\boldsymbol{p}^x, -1) = w(\boldsymbol{p}^x, -1) = w_+ \cdot p^x(-1) \,. \tag{15}$$

Recall that $\mathrm{V}(w) = \frac{w_- w_+}{w_- + w_+}$. Note that $\gamma > 0$ implies $\mathrm{V}(w) > 0$, and thus $w_-, w_+ > 0$. Since $p^x(-1) + p^x(+1) = 1$, having $w(\boldsymbol{p}^x, y) < V(w)$ for both $y = -1$ and $y = +1$ implies the absurd:

$$1 = \frac{w(\boldsymbol{p}^x, +1)}{w_-} + \frac{w(\boldsymbol{p}^x, -1)}{w_+} < \frac{\mathrm{V}(w)}{w_-} + \frac{\mathrm{V}(w)}{w_+} = \frac{w_+}{w_- + w_+} + \frac{w_-}{w_- + w_+} = 1 \,. \tag{16}$$

Together with Equation (13) and the definition of $\widehat{h}_S$ from Algorithm 1, this consequently means that $\widehat{h}_S(x_i) = y_i$ for all $i \in [m]$, concluding the proof. ■

The next lemma shows that, if the size m of the sample in Lemma 13 is large enough, then the hypothesis returned by Algorithm 1 has small generalization error. It is immediate to see that, together with Lemma 13, this implies Theorem 12.

Lemma 14 *Assume the hypotheses of Lemma 13. For any $\epsilon, \delta \in (0, 1)$, if the size m of the sample given to Algorithm 1 satisfies*

$$m \ge \frac{\ln(2/\delta)}{\epsilon} + m_0\left(\frac{\gamma}{3}, \frac{\delta}{2T}\right) \cdot T \cdot \left(1 + \frac{\ln(m)}{\epsilon}\right) \,,$$

then with probability at least $1 - \delta$ the output $\widehat{h}_S$ of Algorithm 1 satisfies $L_{\mathcal{D}}^{w}(\widehat{h}_S) \le \epsilon$. Therefore, Algorithm 1 is a $(w, 0)$-learner for $\mathcal{F}$ with sample complexity $m(\epsilon, \delta) = \widetilde{O}\Big(m_0\big(\frac{\gamma}{3}, \frac{\delta}{2T}\big) \cdot \frac{\ln(1/\delta)}{\epsilon\cdot\gamma^2}\Big)$.

Proof First, we apply Lemma 13 with $\delta/2$ in place of δ; we obtain that, with probability at least $1-\delta/2$, the hypothesis $\widehat{h}_S$ is consistent with the labeled sample S. Next, we apply a standard compression-based generalization argument (see, e.g., Theorem 2.8 from Schapire and Freund (2012)). To this end, note that one can construct a compression scheme for $\widehat{h}_S$ of size κ equal to the total size of the samples on which $\mathcal{A}$ is invoked, that is, $\kappa = \widehat{m} \cdot T$. By Theorem 2.8 from Schapire and Freund (2012) with $\delta/2$ in place of δ we get that, with probability at least $1-\delta/2$,

$$L_{\mathcal{D}}^{w}(\widehat{h}_S) \le \frac{\kappa \ln(m) + \ln(2/\delta)}{m-\kappa} \; . \tag{17}$$

Straightforward calculations show that the right-hand side is at most ϵ whenever:

$$\begin{aligned} m &\ge \frac{\ln(2/\delta)}{\epsilon} + \kappa \cdot \left(1 + \frac{\ln(m)}{\epsilon}\right) && (18)\\ &= \frac{\ln(2/\delta)}{\epsilon} + m_0\left(\frac{\gamma}{3}, \frac{\delta}{2T}\right) \cdot T \cdot \left(1 + \frac{\ln(m)}{\epsilon}\right) \; . && (19) \end{aligned}$$

A union bound completes the first part of the claim, and shows that Algorithm 1 is a $(w,0)$-learner for $\mathcal{F}$. The condition on the sample complexity of Algorithm 1 then follows immediately by definition of $T = O\big(\frac{\ln(m)}{\gamma^2}\big)$. ■

Remark on adaptive boosting. Traditional boosting algorithms, such as the well-known AdaBoost (Schapire and Freund, 2012), do not assume prior knowledge of the margin parameter γ and adapt to it dynamically during execution. In contrast, the boosting algorithm in Algorithm 1, along with our broader approach, requires an initial estimate of γ as input. If this estimate is too large, the algorithm may fail. However, this issue can be addressed through a straightforward binary search procedure: we iteratively adjust our guess, halving it when necessary based on observed outcomes. This adds only a logarithmic overhead of $O(\ln(1/\gamma))$ to the runtime, without affecting sample complexity bounds.

We conclude this subsection with a proof of the second part of Theorem A.

Lemma 15 *Let $\mathcal{X}$ be any domain, let $\mathcal{Y} = \{-1,+1\}$, and let $w = (w_+, w_-) \in (0,1]^2$ be a cost. If $z \ge \mathrm{V}(w)$ then there is a random guess h such that the algorithm that always returns h is a (w,z)-learner for every $\mathcal{F} \subseteq \mathcal{Y}^{\mathcal{X}}$. As a consequence, for some domain $\mathcal{X}$ and some $\mathcal{F} \subseteq \mathcal{Y}^{\mathcal{X}}$, the said learner cannot be boosted to a (w,z')-learner for any $z' < \mathrm{V}(w)$.*

Proof By definition of $\mathrm{V}(w)$ there exists a distribution $\boldsymbol{p} \in \Delta_{\mathcal{Y}}$ such that $w(\boldsymbol{p},\boldsymbol{q}) \le \mathrm{V}(w)$ for every $\boldsymbol{q} \in \Delta_{\mathcal{Y}}$ over $\mathcal{Y}$. Consider then the algorithm that ignores the input and returns the randomized hypothesis h that, when invoked, returns a label distributed independently according to $\boldsymbol{p}$. Clearly, for every distribution $\mathcal{D}$ over $\mathcal{X}$,

$$L_{\mathcal{D}}^{w}(h) \le w(\boldsymbol{p},\boldsymbol{q}) \le \mathrm{V}(w) \; , \tag{20}$$

where $\boldsymbol{q}$ is the marginal of $\mathcal{D}$ over $\mathcal{Y}$. This proves that the algorithm is a (w,z)-learner for every $\mathcal{F} \subseteq \mathcal{Y}^{\mathcal{X}}$. Let us now show that for some $\mathcal{F} \subseteq \mathcal{Y}^{\mathcal{X}}$ cannot be boosted to a (w,z')-learner for any $z' < \mathrm{V}(w)$. If $\mathrm{V}(w) = 0$ then this is trivial, as $w \ge 0$; thus we can assume $\mathrm{V}(w) > 0$. Suppose then that a (w,z')-learner for $\mathcal{F}$ exists, where $z' < \mathrm{V}(w)$. By item (1), it follows that a $(w,0)$-learner for

$\mathcal{F}$ exists. Recall that $\mathrm{V}(w) = \frac{w_- w_+}{w_- + w_+}$. Since we assumed $\mathrm{V}(w) > 0$, it follows that $w_-, w_+ > 0$. Then, a $(w, 0)$-learner is a strong learner in the PAC sense. Now choose $\mathcal{F}$ to be any non-learnable class; for instance, let $\mathcal{F} = \mathcal{Y}^{\mathbb{N}}$ (see Shalev-Shwartz and Ben-David (2014, Theorem 5.1)). Then, no strong PAC learner and therefore no $(w, 0)$-learner exists for $\mathcal{F}$. Hence a (w, z')-learner for $\mathcal{F}$ with $z' < \mathrm{V}(w)$ does not exist, too. ■

A.2. Multi-objective losses: boosting a $(\boldsymbol{w}, \boldsymbol{z})$-learner

This section considers the boosting problem for learners that satisfy multi-objective costs, as captured by the notion of $(\boldsymbol{w}, \boldsymbol{z})$-learners (Definition 3). Our main question is then:

When can a $(\boldsymbol{w}, \boldsymbol{z})$-learner be boosted to a $(\boldsymbol{w}, \boldsymbol{0})$-learner?

For the scalar case, we showed in Appendix A.1 that the threshold to boost (w, z)-learners is the value of the game $\mathrm{V}(w)$, which is the best bound on w one can achieve by just tossing a coin—independently of the example x at hand, and even of the specific distribution $\mathcal{D}$ over $\mathcal{X}$. One may thus guess that the same characterization holds for boosting $(\boldsymbol{w}, \boldsymbol{z})$-learners. It turns out that this guess fails, as we show below. We thus adopt a different and stronger notion of "trivial learner", obtained by exchanging the order of the players. It turns out that this is the right notion, in the sense that it is equivalent to being non-boostable.

For the sake of simplicity we assume a specific $\boldsymbol{w}$ that we call *population-driven* cost. This will help conveying the key ideas, without having to deal with the technical details of the more general version of our results (Appendix C). The population-driven cost is the one that counts, separately, false negatives and false positives. More precisely, $\boldsymbol{w} = (w_-, w_+)$ where w_- and w_+ are now two cost functions that for all $i, j \in \mathcal{Y}$ satisfy:

$$w_-(i, j) = \mathbb{I}\{i = -1 \wedge j = +1\} \ , \tag{21}$$

$$w_+(i, j) = \mathbb{I}\{i = +1 \wedge j = -1\} \ . \tag{22}$$

For $\boldsymbol{z} = (z_-, z_+) \in \mathbb{R}^2_{\geq 0}$, a $(\boldsymbol{w}, \boldsymbol{z})$-learner then ensures simultaneously a false-negative rate arbitrarily close to z_- and a false-positive rate arbitrarily close to z_+.

A.2.1. COIN ATTAINABILITY

Let $\boldsymbol{w}$ be as above, and let $\boldsymbol{z} \in \mathbb{R}^2_{\geq 0}$. As a first attempt at answering our question above, we guess that $\boldsymbol{z}$ is non-boostable when it is attainable by a random coin toss in the same way as for scalar costs (Appendix A.1). That is, we say that $\boldsymbol{z}$ is *trivially attainable* w.r.t. $\boldsymbol{w}$ if there exists $\boldsymbol{p} \in \Delta_{\mathcal{Y}}$ such that for every $\boldsymbol{q} \in \Delta_{\mathcal{Y}}$ we have $\boldsymbol{w}(\boldsymbol{p}, \boldsymbol{q}) \leq \boldsymbol{z}$. Clearly, for any such $\boldsymbol{z}$ there would exist a $(\boldsymbol{w}, \boldsymbol{z})$-learner that is not boostable for some class $\mathcal{F}$. If we could also prove the converse, that any $\boldsymbol{z}$ *not* trivially attainable is boostable, then we would have determined that trivial attainability is equivalent non-boostability.

Unfortunately, our guess fails. To see why, fix any distribution $\boldsymbol{p} = (p_-, p_+)$. Define then $\boldsymbol{q} = (q_-, q_+)$ as follows: if $p_- \geq 1/2$ then $q_- = 0$, otherwise $q_+ = 0$. It follows immediately that either $p_-(1 - q_-) \geq 1/2$ or $p_+(1 - q_+) \geq 1/2$. That is, $\boldsymbol{w}(\boldsymbol{p}, \boldsymbol{q})$ is at least $1/2$ in at least one coordinate. As a consequence, no $\boldsymbol{z} < (1/2, 1/2)$ is trivially attainable. Thus, if our guess is correct, it should be the case that any $(\boldsymbol{w}, \boldsymbol{z})$-learner with $\boldsymbol{z} < (1/2, 1/2)$ is boostable.

We can easily show that that is not the case. Let $\boldsymbol{z} = (1/4, 1/4)$ and consider the following $(\boldsymbol{w}, \boldsymbol{z})$-learner. Upon receiving a sufficiently large labeled sample from some distribution $\mathcal{D}$ over $\mathcal{X}$, the learner computes an estimate $\hat{\boldsymbol{q}} = (\hat{q}_-, \hat{q}_+)$ of the marginal of $\mathcal{D}$ over $\mathcal{Y}$. The learner then computes $\boldsymbol{p} = (p_-, p_+)$ that satisfies:

$$p_- \cdot (1 - \hat{q}_-) \leq \frac{1}{4} \quad \text{and} \quad p_+ \cdot (1 - \hat{q}_+) \leq \frac{1}{4} \, . \tag{23}$$

It is not hard to show that such a $\boldsymbol{p}$ always exists. The learner thus outputs h such that $h(x) = \boldsymbol{p}$ for all $x \in \mathcal{X}$ independently. As the number of samples received by the learner increases, the estimate $\hat{\boldsymbol{q}}$ approaches $\boldsymbol{q}$, and thus the loss of the learner approaches $\boldsymbol{z}$. Clearly, however, such a learner cannot be boosted, as its predictions are independent of the example x at hand. We conclude that there exist $\boldsymbol{z}$ that are not trivially attainable and yet are not boostable.

The example above suggest a fix to our notion of "non-boostable learner". The fix consists in considering again prediction by a coin $\boldsymbol{p}$, where, however, we allow $\boldsymbol{p}$ to be a function of $\boldsymbol{q}$. This leads to the following definition:

Definition 16 *Let $\boldsymbol{w} = (w_-, w_+)$ be the population-driven cost and let $\boldsymbol{z} = (z_-, z_+) \in \mathbb{R}^2_{\geq 0}$. Then, $\boldsymbol{z}$ is* coin-attainable *w.r.t. $\boldsymbol{w}$ if for every distribution $\boldsymbol{q} = (q_-, q_+)$ there exists a distribution $\boldsymbol{p} = (p_-, p_+)$ such that:*

$$w_-(\boldsymbol{p}, \boldsymbol{q}) = p_- \cdot (1 - q_-) \leq z_- \quad \textit{and} \quad w_+(\boldsymbol{p}, \boldsymbol{q}) = p_+ \cdot (1 - q_+) \leq z_+ \, .$$

Note how Definition 16 mirrors the game of Appendix A.1, but with a reverted *order of the players*: here, the predictor plays second. It should be clear that, if $\boldsymbol{z}$ is coin-attainable, then there exists a $(\boldsymbol{w}, \boldsymbol{z})$-learner that is not boostable. That is the learner described above, which gets an estimate $\hat{\boldsymbol{q}}$ of $\boldsymbol{q}$ from the sample, and then returns h such that $h(x) = \boldsymbol{p}$, where $\boldsymbol{p}$ satisfies the condition in Definition 16 with respect to $\hat{\boldsymbol{q}}$. What is not clear is whether the converse holds, too: is a $(\boldsymbol{w}, \boldsymbol{z})$-learner boostable whenever $\boldsymbol{z}$ is *not* coin-attainable? It turns out that this is the case, as the next subsection shows.

A.2.2. COIN-ATTAINABILITY EQUALS NON-BOOSTABILITY

Let us try to boost a $(\boldsymbol{w}, \boldsymbol{z})$-learner $\mathcal{A}$. We can do so by reduction to the scalar case of Appendix A.1. To this end choose any $\boldsymbol{\alpha} = (\alpha_1, \alpha_2) \in \Delta_{\mathcal{Y}}$. Define $\boldsymbol{\alpha} \cdot \boldsymbol{w} : \mathcal{Y}^2 \to \mathbb{R}_{\geq 0}$ by:

$$(\boldsymbol{\alpha} \cdot \boldsymbol{w})(i, j) = \alpha_1 \cdot w_-(i, j) + \alpha_2 \cdot w_+(i, j) \qquad \forall i, j \in \mathcal{Y} \, . \tag{24}$$

For conciseness we shall write $w = \boldsymbol{\alpha} \cdot \boldsymbol{w}$. In words, w is the convex combination of w_-, w_+ according to $\boldsymbol{\alpha}$, and therefore $w = \left(\begin{smallmatrix} 0 & \alpha_1 \\ \alpha_2 & 0 \end{smallmatrix}\right)$. Therefore $w : \mathcal{Y}^2 \to [0, 1]$ and $w(i, i) = 0$ for each $i \in \mathcal{Y}$. Now let $z = \boldsymbol{\alpha} \cdot \boldsymbol{z}$. Now we make the following crucial observation: since $\mathcal{A}$ is a $(\boldsymbol{w}, \boldsymbol{z})$-learner, then it is also a (w, z)-learner. This is immediate to see from the definition of $(\boldsymbol{w}, \boldsymbol{z})$ and the definition of w. It follows that, by the results of Appendix A.1, $\mathcal{A}$ can be boosted if $z < \mathrm{V}(w)$. Summarizing, we can boost $\mathcal{A}$ whenever:

$$\boldsymbol{\alpha} \cdot \boldsymbol{z} < \mathrm{V}(\boldsymbol{\alpha} \cdot \boldsymbol{w}) \, . \tag{25}$$

That is, if there is *any* $\boldsymbol{\alpha} \in \Delta_{\mathcal{Y}}$ satisfying Equation (25), then $\mathcal{A}$ can be boosted to a strong learner. Moreover, if that is the case, then one can prove that one can ensure $\boldsymbol{\alpha} > \boldsymbol{0}$. Then, one can easily see that $\mathcal{A}^*$ is a $(\boldsymbol{w}, \boldsymbol{0})$-learner. Our main question thus becomes:

Is it true that, if $\boldsymbol{z}$ is not coin-attainable, then there is $\boldsymbol{\alpha} \in \Delta_{\mathcal{Y}}$ such that $\boldsymbol{\alpha} \cdot \boldsymbol{z} < \mathrm{V}(\boldsymbol{\alpha} \cdot \boldsymbol{w})$?

We show that this is indeed the case, and therefore $\boldsymbol{z}$ is coin-attainable if and only if $\boldsymbol{\alpha} \cdot \boldsymbol{z} \geq \mathrm{V}(\boldsymbol{\alpha} \cdot \boldsymbol{w})$ for all $\boldsymbol{\alpha} \in \Delta_{\mathcal{Y}}$. Equivalently, all $(\boldsymbol{w}, \boldsymbol{z})$-learners are boostable if and only if $\boldsymbol{\alpha} \cdot \boldsymbol{z} < \mathrm{V}(\boldsymbol{\alpha} \cdot \boldsymbol{w})$ for some $\boldsymbol{\alpha} \in \Delta_{\mathcal{Y}}$. As a consequence, every $\boldsymbol{z}$ is either coin-attainable or boostable. Formally, we have:

Theorem 17 *Let $\mathcal{Y} = \{-1, +1\}$, let $\boldsymbol{w}$ be the population-driven cost, and let $\boldsymbol{z} = (z_-, z_+) \in \mathbb{R}^2_{\geq 0} \setminus \mathbf{0}$. Then $\boldsymbol{z}$ is coin-attainable w.r.t. $\boldsymbol{w}$ if and only if $\boldsymbol{\alpha} \cdot \boldsymbol{z} \geq \mathrm{V}(\boldsymbol{\alpha} \cdot \boldsymbol{w})$, where:*

$$\boldsymbol{\alpha} = \left(\frac{\sqrt{z_+}}{\sqrt{z_-} + \sqrt{z_+}}, \frac{\sqrt{z_-}}{\sqrt{z_-} + \sqrt{z_+}} \right) \in \Delta_{\mathcal{Y}} \,. \tag{26}$$

Equivalently, $\boldsymbol{z}$ is coin-attainable w.r.t. $\boldsymbol{w}$ if and only if $\sqrt{z_-} + \sqrt{z_+} \geq 1$.

Proof First of all, by straightforward calculations we determine:

$$\boldsymbol{\alpha} \cdot \boldsymbol{z} = \sqrt{z_-}\sqrt{z_+} \,, \tag{27}$$

$$\mathrm{V}\left(\boldsymbol{\alpha} \cdot \boldsymbol{w}\right) = \mathrm{V}\left(\begin{smallmatrix} 0 & \alpha_1 \\ \alpha_2 & 0 \end{smallmatrix} \right) = \frac{\sqrt{z_-}\sqrt{z_+}}{\sqrt{z_-} + \sqrt{z_+}} \,. \tag{28}$$

This proves that $\boldsymbol{\alpha} \cdot \boldsymbol{z} \geq \mathrm{V}(\boldsymbol{\alpha} \cdot \boldsymbol{w})$ if and only if $\sqrt{z_-} + \sqrt{z_+} \geq 1$.

Now, for the “only if” direction, by von Neumann’s Minimax Theorem and by definition of $\boldsymbol{\alpha}$,

$$V(\boldsymbol{\alpha} \cdot \boldsymbol{w}) = \max_{\boldsymbol{q} \in \Delta_{\mathcal{Y}}} \min_{\boldsymbol{p} \in \Delta_{\mathcal{Y}}} \frac{\sqrt{z_+}}{\sqrt{z_-} + \sqrt{z_+}} \cdot p_- \cdot (1 - q_-) + \frac{\sqrt{z_-}}{\sqrt{z_-} + \sqrt{z_+}} \cdot p_+ \cdot (1 - q_+) \tag{29}$$

$$\leq \frac{\sqrt{z_+} \cdot z_-}{\sqrt{z_-} + \sqrt{z_+}} + \frac{\sqrt{z_-} \cdot z_+}{\sqrt{z_-} + \sqrt{z_+}} = \sqrt{z_-}\sqrt{z_+} = \boldsymbol{\alpha} \cdot \boldsymbol{z} \,, \tag{30}$$

where the inequality holds as $\boldsymbol{z}$ is coin-attainable.

For the “if” direction, suppose $\boldsymbol{\alpha} \cdot \boldsymbol{z} \geq \mathrm{V}(\boldsymbol{\alpha} \cdot \boldsymbol{w})$, hence $\sqrt{z_-} + \sqrt{z_+} \geq 1$. We claim that this implies $\boldsymbol{z}$ is coin-attainable. Fix indeed any $\boldsymbol{q} = (q_-, q_+) \in \Delta_{\mathcal{Y}}$. We consider three cases: (1) $q_+ \leq z_-$, (2) $q_- \leq z_+$, and (3) $q_+ > z_-$ and $q_- > z_+$. If $q_+ \leq z_-$ then choose $\boldsymbol{p} = (1, 0)$, otherwise if $q_- \leq z_+$ choose $\boldsymbol{p} = (0, 1)$. It is immediate to see that this implies $p_- \cdot (1 - q_-) \leq z_-$ and $p_+ \cdot (1 - q_+) \leq z_+$. Suppose then $q_+ > z_-$ and $q_- > z_+$, which implies $\boldsymbol{q} > \mathbf{0}$. We claim that:

$$\frac{z_-}{q_+} + \frac{z_+}{q_-} \geq 1 \,. \tag{31}$$

Indeed, letting $a = z_-$, $b = z_+$, $x = q_+$, multiplying both sides by $x(1-x)$, and rearranging yields:

$$a \cdot (1 - x) + b \cdot x - x(1 - x) \geq 0 \,, \tag{32}$$

which can be easily checked to hold for all x whenever $\sqrt{a} + \sqrt{b} \geq 1$. Equation (31) then implies the existence of p_-, p_+ such that $p_- + p_+ = 1$ and that:

$$p_- \cdot (1 - q_-) \leq \frac{z_-}{q_+} \cdot (1 - q_-) = z_- \,, \tag{33}$$

$$p_+ \cdot (1 - q_+) \leq \frac{z_+}{q_-} \cdot (1 - q_+) = z_+ \,. \tag{34}$$

We conclude that $\boldsymbol{z}$ is coin-attainable. ■

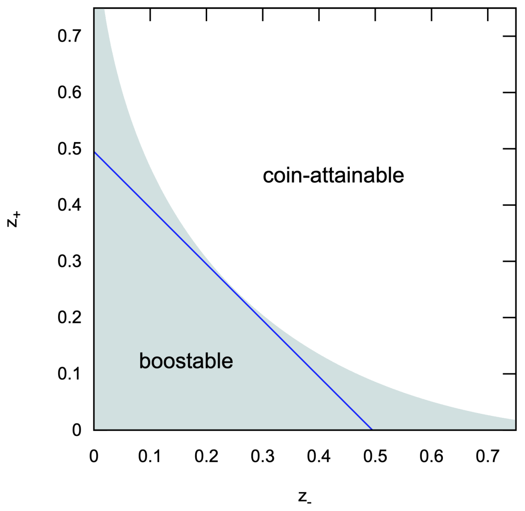

Figure 3: Duality in a picture. For a multi-objective cost $\boldsymbol{w} : \mathcal{Y}^2 \to \mathbb{R}^2_{\geq 0}$, the set $C(\boldsymbol{w})$ of all coin-attainable vectors $\boldsymbol{z}$ is the intersection of all halfspaces in the form $H(\boldsymbol{\alpha}) = \{\boldsymbol{x} \in \mathbb{R}^2 : \boldsymbol{\alpha} \cdot \boldsymbol{x} \geq \mathrm{V}(\boldsymbol{\alpha} \cdot \boldsymbol{w})\}$. As a consequence, the complement of $C(\boldsymbol{w})$ is the set of all boostable vectors $\boldsymbol{z}$. This is a special case of a more general result, stated in Appendix C, that holds for any $k \geq 2$ and any multi-objective cost $\boldsymbol{w} : \mathcal{Y}^2 \to \mathbb{R}^r_{\geq 0}$. The straight blue line is the boundary of a specific $H(\boldsymbol{\alpha})$: for any $\boldsymbol{z}$ in it, every $(\boldsymbol{w}, \boldsymbol{z})$-learner is boostable w.r.t. the cost $\boldsymbol{\alpha} \cdot \boldsymbol{w}$.

A.3. A duality, and a geometric interpretation

Theorem 17 has an interesting geometric interpretation. In fact, one can easily check that for every $\boldsymbol{z} \in \mathbb{R}^2_{\geq 0}$ it holds that $\sqrt{z_-} + \sqrt{z_+} \geq 1$ if and only if $\boldsymbol{\alpha} \cdot \boldsymbol{z} \geq \mathrm{V}(\boldsymbol{\alpha} \cdot \boldsymbol{w})$ for every $\boldsymbol{\alpha} \in \Delta_{\mathcal{Y}}$, and not just for the single $\boldsymbol{\alpha}$ specified in Theorem 17. That is, we have:

Theorem 18 *Let $\mathcal{Y} = \{-1, +1\}$, let $\boldsymbol{w}$ be the population-driven cost, and let $\boldsymbol{z} \in \mathbb{R}^2_{\geq 0}$. Then $\boldsymbol{z}$ is coin-attainable w.r.t. $\boldsymbol{w}$ if and only if $\boldsymbol{\alpha} \cdot \boldsymbol{z} \geq \mathrm{V}(\boldsymbol{\alpha} \cdot \boldsymbol{w})$ for every $\boldsymbol{\alpha} \in \Delta_{\mathcal{Y}}$.*

Theorem 18 says that $\boldsymbol{z}$ is coin-attainable if and only if there is no way to "scalarize" $\boldsymbol{w}$ so as to obtain a boostable learner w.r.t. the corresponding scalarized cost z. From a geometric perspective, the result can be read as follows. For any $\boldsymbol{\alpha} \in \Delta_{\mathcal{Y}}$ consider the following halfspace:

$$H(\boldsymbol{\alpha}) = \{\boldsymbol{x} \in \mathbb{R}^2 \, : \, \boldsymbol{\alpha} \cdot \boldsymbol{x} \geq \mathrm{V}(\boldsymbol{\alpha} \cdot \boldsymbol{w})\} \; . \tag{35}$$

Let moreover $C(\boldsymbol{w})$ be the set of all $\boldsymbol{z}$ that are coin-attainable w.r.t. $\boldsymbol{w}$. Then Theorem 18 says:

$$C(\boldsymbol{w}) = \bigcap_{\boldsymbol{\alpha} \in \Delta_{\mathcal{Y}}} H(\boldsymbol{\alpha}) \; . \tag{36}$$

In other words, the set of coin-attainable vectors is precisely the intersection of all halfspaces $H(\boldsymbol{\alpha})$. It follows that $C(\boldsymbol{w})$ is convex: and, indeed, if two vectors $\boldsymbol{z}$ and $\boldsymbol{z}'$ are both coin-attainable then it is easy to see that $a \cdot \boldsymbol{z} + (1-a) \cdot \boldsymbol{z}'$ is coin-attainable, too, for every $a \in [0,1]$. Figure 3 gives a pictorial description of this geometric interpretation. The proof of Theorem 18 is not given here, as it is a special case of Theorem 8.

We continue this subsection we providing a few more claims regrading the geometric interpretation of the results.

Lemma 19 *Let $\boldsymbol{w} = (w_1, \dots, w_r)$ as above. Assume all $\boldsymbol{z} \in C(\boldsymbol{w})$ are not redundant for $\boldsymbol{w}$. Denote the canonical coin region as $C = \{\boldsymbol{z} \in [0,1]^2 : \sqrt{z_+} + \sqrt{z_-} \geq 1\}$. Then, for any $\boldsymbol{z} \in [0,1]^r$:*

$$\boldsymbol{z} \in C(\boldsymbol{w}) \implies \forall i \in [r], \;\; z_i \geq V(w_i) \; .$$

Proof Fix any $i \in [r]$. By Section 3, we know that $V(w_i) = \frac{w_+^i w_-^i}{w_+^i + w_-^i}$, and by assumption we have :

$$\langle \boldsymbol{z}_c, w_i \rangle = z_{+_c} \cdot w_+^i + z_{-_c} \cdot w_-^i \leq z_i \; .$$

By applying Claim 20 with $(a,b) = \boldsymbol{z}_c$, $(x,y) = w_i$, we also get $\langle \boldsymbol{z}_c, w_i \rangle \geq V(w_i)$, as needed. ■

Claim 20 *Let $a, b \geq 0$ such that $\sqrt{a} + \sqrt{b} \geq 1$, and let $x, y \geq 0$. Then: $ax + by \geq \frac{xy}{x+y}$.*

Proof It suffices to show that $ax^2 + (a+b-1)xy + by^2 \geq 0$. Set $c = \sqrt{a}$ and $d = \sqrt{b}$, and substitute in the above LHS to get:

$$c^2x^2 + d^2y^2 + \big(c^2 + d^2 - 1\big)\, x\, y = (cx - dy)^2 + \big((c+d)^2 - 1\big)xy \geq 0 \; ,$$

where the inequality indeed holds since all parameters are non-negative and $c + d \geq 1$. ■

Observation 21 *Let $\boldsymbol{w} = (w_1, \dots, w_r)$, and let $\boldsymbol{z} \in [0,1]^r$. Then, the feasible region set $R(\boldsymbol{w}, \boldsymbol{z})$ is obtained by the intersection of halfplanes of the form $\langle e, w_i \rangle = z_i$, and is therefore a convex polygon.*

A.4. Proof of Theorem B

Proof of second item of Theorem B. We show that, if $\boldsymbol{z} \in C(\boldsymbol{w})$, then there is a trivial learner $\mathcal{A}$ that is a $(\boldsymbol{w}, \boldsymbol{z})$-learner for every $\mathcal{F} \subseteq \mathcal{Y}^{\mathcal{X}}$. By choosing $\mathcal{F}$ to be any non-learnable class this implies that $\mathcal{A}$ is not boostable to a $(\boldsymbol{w}, \boldsymbol{z}')$-learner for $\mathcal{F}$ for any $\boldsymbol{z}' \notin C(\boldsymbol{w})$, for otherwise the first item above would imply the existence of a $(\boldsymbol{w}, \boldsymbol{0})$-learner for $\mathcal{F}$. The learner $\mathcal{A}$ works as follows. Given a sample S of m examples drawn i.i.d. from $\mathcal{D}$ and labeled by any $f : \mathcal{X} \to \mathcal{Y}$, it estimates the marginal $\boldsymbol{q}$ of $\mathcal{D}$ on $\mathcal{Y}$. Denote that estimate by $\hat{\boldsymbol{q}}$. It is well known that $\mathcal{A}$ can ensure $\hat{\boldsymbol{q}}$ is within total variation distance ϵ of $\boldsymbol{q}$ with probability at least $1 - \delta$ by taking $m = \Theta\big(\frac{k + \ln(1/\delta)}{\epsilon^2}\big)$, see for instance (Canonne, 2020, Theorem 1). Assume then this is the case. Let $\boldsymbol{p} \in \Delta$ be a distribution such that $\boldsymbol{w}(\boldsymbol{p}, \hat{\boldsymbol{q}}) \leq \boldsymbol{z}$; note that $\boldsymbol{p}$ exists by definition of coin attainability. Then, $\mathcal{A}$ returns the hypothesis $h : \mathcal{X} \to \Delta_{\mathcal{Y}}$ such that $h(x) = \boldsymbol{p}$ for all $x \in \mathcal{X}$. Since $\boldsymbol{w} \in [0,1]^r$, then

$$L_{\mathcal{D}}^{\boldsymbol{w}}(h) \leq \boldsymbol{w}(\boldsymbol{q}, \hat{\boldsymbol{q}}) + \epsilon \cdot \boldsymbol{1} = \boldsymbol{z} + \epsilon \cdot \boldsymbol{1} \; . \tag{37}$$

This proves that $\mathcal{A}$ is a $(\boldsymbol{w}, \boldsymbol{z})$-learner. ■

A.4.1. BOOSTING TO ARBITRARY COSTS

Next, we prove that for every $\mathcal{F} \subseteq \mathcal{Y}^{\mathcal{X}}$, any $(\boldsymbol{w}, \boldsymbol{z})$-learner for $\mathcal{F}$ that is boostable, is also boostable for for every other cost. This is a strengthening of the first item of Theorem B, and is formally stated below.

Theorem 22 *Let $\mathcal{Y} = \{-1, 1\}$, let $\boldsymbol{w} = (w_1, \dots, w_r)$ where $w_i : \mathcal{Y}^2 \to [0, 1]$ is a cost, and let $\boldsymbol{z} \in [0, 1]^r$. Then, for every $\mathcal{F} \subseteq \mathcal{Y}^{\mathcal{X}}$, every boostable $(\boldsymbol{w}, \boldsymbol{z})$-learner can be boosted to a $(\boldsymbol{w}', \mathbf{0})$-learner for any other cost vector $\boldsymbol{w}' = (w'_1, \dots, w'_{r'})$.*

Proof By assumption, every $(e_+, e_-) \in R(\boldsymbol{w}, \boldsymbol{z})$ satisfies $\sqrt{e_+} + \sqrt{e_-} < 1$. That is, we have $R(\boldsymbol{w}, \boldsymbol{z}) \cap C = \emptyset$, where $C = \{\boldsymbol{e} \in [0, 1]^2 : \sqrt{e_+} + \sqrt{e_-} \geq 1\}$. Notice that both sets $R(\boldsymbol{w}, \boldsymbol{z})$ and C are compact and convex. Thus, by the hyperplane separation theorem there exist a nonzero vector $u \in [0, 1]^2$ and a real number $a \geq 0$ such that $\langle \boldsymbol{e}, u \rangle \leq a$ for every $\boldsymbol{e} \in R(\boldsymbol{w}, \boldsymbol{z})$ and $\langle \boldsymbol{z}_c, u \rangle > a$ for every $\boldsymbol{z}_c \in C$. In particular, this implies that any $(\boldsymbol{w}, \boldsymbol{z})$-learner (multi-objective) is also a (u, a)-learner (cost-sensitive), and that $a < V(u)$. By Theorem 12, any such learner can be boosted to a $(u, 0)$-learner. Then, by Claims 24 and 25, we get that any such learner is also a $(\boldsymbol{w}', \mathbf{0})$-learner for *any* cost vector $\boldsymbol{w}' = (w'_1, \dots, w'_{r'})$. ■

Definition 23 *Let $\mathcal{Y} = \{-1, 1\}$. Let $\boldsymbol{w}^*$ the canonical cost in the binary case: $\boldsymbol{w}^* = (w_p, w_n)$ with $w_p = \left(\begin{smallmatrix} 0 & 0 \\ 1 & 0 \end{smallmatrix}\right)$ and $w_n = \left(\begin{smallmatrix} 0 & 1 \\ 0 & 0 \end{smallmatrix}\right)$. That is, w_p counts the false positives, w_n the false negatives.*

Claim 24 *Let $\mathcal{Y} = \{-1, 1\}$, let $w : \mathcal{Y}^2 \to [0, 1]$ be any cost such that $w = \left(\begin{smallmatrix} 0 & a_- \\ a_+ & 0 \end{smallmatrix}\right)$ where $a_-, a_+ > 0$, and let $\boldsymbol{w}^*$ denote the canonical cost in the binary case. Then, for every $\mathcal{F} \subseteq \mathcal{Y}^{\mathcal{X}}$, any $(w, 0)$-learner for $\mathcal{F}$ is also a $(\boldsymbol{w}^*, \mathbf{0})$-learner $\mathcal{F}$.*

Proof Let $\mathcal{A}$ denote the $(w, 0)$-learner for $\mathcal{F}$, and let $m_0(\cdot, \cdot)$ denote its sample complexity. Let $f \in \mathcal{F}$, distribution $\mathcal{D}$, and $\epsilon, \delta \in (0, 1)$. Then, by definition of a cost-sensitive learner, the following holds. If S is a sample of $m_0(\epsilon, \delta)$ examples drawn i.i.d. from $\mathcal{D}$ and labeled by f, then $\mathcal{A}(S)$ returns a predictor $h : \mathcal{X} \to \Delta_{\mathcal{Y}}$ such that with probability at least $1 - \delta$, $L^w_{\mathcal{D}}(h) \leq \epsilon$. By the definition of the loss we get:

$$L^w_{\mathcal{D}}(h) = a_- \cdot \mathbb{P}_{x \sim \mathcal{D}, Y \sim h(x)} \Big[Y = -1 \wedge f(x) = 1 \Big] + a_+ \cdot \mathbb{P}_{x \sim \mathcal{D}, Y \sim h(x)} \Big[Y = 1 \wedge f(x) = -1 \Big] \leq \epsilon$$

In particular, we have both:

$$\mathbb{P}_{x \sim \mathcal{D}, Y \sim h(x)} \Big[Y = 1 \wedge f(x) = -1 \Big] \leq \epsilon / a_+, \text{ and } \mathbb{P}_{x \sim \mathcal{D}, Y \sim h(x)} \Big[Y = -1 \wedge f(x) = 1 \Big] \leq \epsilon / a_-.$$

For any $\epsilon_1, \delta_1 > 0$, let $m_1(\epsilon_1, \delta_1) = m_0(\epsilon_1 \cdot \min\{a_-, a_+\}, \delta_1)$. Then, we get that $\mathcal{A}$ is in fact a $(\boldsymbol{w}^*, \mathbf{0})$-learner for $\mathcal{F}$ with sample complexity m_1. ■

Claim 25 *Let $\mathcal{Y} = \{-1, 1\}$, let $w : \mathcal{Y}^2 \to [0, 1]$ be any cost, and let $\boldsymbol{w}^*$ denote the canonical cost in the binary case. Let $\boldsymbol{w} = (w_1, \dots, w_r)$ be any cost vector. Then, for every $\mathcal{F} \subseteq \mathcal{Y}^{\mathcal{X}}$, any $(\boldsymbol{w}^*, \mathbf{0})$-learner for $\mathcal{F}$ is also a $(\boldsymbol{w}, \mathbf{0})$-learner for $\mathcal{F}$.*

Proof Denote the cost w_i in a matrix form as $w_i = \begin{pmatrix} 0 & a_-^i \\ a_+^i & 0 \end{pmatrix}$ where $a_-^i, a_+^i \in [0,1]$. Let $f \in \mathcal{F}$, distribution $\mathcal{D}$, and $\epsilon \in (0,1)$. Consider a predictor $h : \mathcal{X} \to \Delta_{\mathcal{Y}}$ such that $L_{\mathcal{D}}^{w_p}(h) \leq \epsilon$ and $L_{\mathcal{D}}^{w_n}(h) \leq \epsilon$. Then, it also holds that for every $i = 1, \dots, r$ we have both:

$$\mathbb{P}_{x\sim\mathcal{D},\,Y\sim h(x)}\Big[Y = 1 \wedge f(x) = -1\Big] \leq \epsilon/a_+^i \quad \text{and} \quad \mathbb{P}_{x\sim\mathcal{D},\,Y\sim h(x)}\Big[Y = -1 \wedge f(x) = 1\Big] \leq \epsilon/a_-^i \ .$$

Thus, we also have:

$$L_{\mathcal{D}}^{w_i}(h) = a_-^i \cdot \mathbb{P}_{x\sim\mathcal{D},\,Y\sim h(x)}\Big[Y = -1 \wedge f(x) = 1\Big] + a_+^i \cdot \mathbb{P}_{x\sim\mathcal{D},\,Y\sim h(x)}\Big[Y = 1 \wedge f(x) = -1\Big] \leq \epsilon \ .$$

■

Appendix B. Equivalence: Cost-Sensitive and Multi-Objective

B.1. Proof of Theorem 7

Algorithm 2 Boosting $(w_{\boldsymbol{\alpha}}, z_{\boldsymbol{\alpha}})$-learners

```
Input: Sample $S = (x_j, y_j)_{j=1}^m$; $(w_{\boldsymbol{\alpha}}, z_{\boldsymbol{\alpha}})$-learners $\mathcal{A}_{\boldsymbol{\alpha}}$ for every $\boldsymbol{\alpha} \in \Delta_r$; parameters $T, \eta, \widehat{m}$
1: Initialize: $a_1(i) = 1$ for all $i = 1, \dots, r$, set $U$ to be the uniform distribution over $[m]$.
2: Define $M(h, i) = \frac{1}{m}\sum_{j=1}^m w_i(h(x_j), y_j) - z_i$ for all $i = 1, \dots, r$ and $h : \mathcal{X} \to \Delta_{\mathcal{Y}}$.
3: for $t = 1, \dots, T$ do
4:     Compute the distribution $\boldsymbol{\alpha}_t \triangleq \frac{\boldsymbol{a}_t}{\sum_i a_t(i)}$ over $[r]$.
5:     Draw a set $S_t$ of $\widehat{m}$ labeled examples i.i.d. from $U$ and obtain $h_t = \mathcal{A}_{\boldsymbol{\alpha}_t}(S_t)$.
6:     For every $i = 1, \dots, r$ let:
               $a_{t+1}(i) \triangleq a_t(i) \cdot e^{\eta \cdot M(h_t, i)}$ .
7: end for
8: return The set of weak predictors $h_1, \dots, h_T$.
```

The direction $1 \implies 2$ is trivial: any $(\boldsymbol{w}, \boldsymbol{z})$-learner is in particular a $(w_{\boldsymbol{\alpha}}, z_{\boldsymbol{\alpha}})$-learner for every $\boldsymbol{\alpha} \in \Delta_r$.

Let us prove $2 \implies 1$. For any $f \in \mathcal{F}$ and every distribution $\mathcal{D}$ over $\mathcal{X}$, let S be a set of m examples sampled i.i.d. from $\mathcal{D}$ and labeled by f, where m will be determined later. Let $\epsilon, \delta > 0$. Then, applying Algorithm 2 over S with access to $\mathcal{A}_{\boldsymbol{\alpha}}$ learners, and with $T = \frac{10\ln(r)}{\epsilon^2}$, $\eta = \sqrt{\frac{2\ln(r)}{T}}$, $\widehat{m} = m_0(\frac{\epsilon}{10}, \frac{\delta}{2rT})$ (where m_0 is the sample complexity of $\mathcal{A}_{\boldsymbol{\alpha}}$) yields the following. Fix any $i \in [r]$. Given that $\max_{t,j} M(h_t, j) \leq 1$ and that $\eta \leq 1$, the standard analysis of Hedge (Freund and Schapire, 1997) shows that:

$$\sum_{t=1}^{T} M(h_t, i) \leq \frac{\ln(r)}{\eta} + \frac{\eta}{2}T + \sum_{t=1}^{T} \mathbb{E}_{i'\sim\boldsymbol{\alpha}_t}\Big[M(h_t, i')\Big] \ . \tag{38}$$

By the guarantee of $\mathcal{A}_{\boldsymbol{\alpha}}$ and the choice of $\widehat{m}$, and by union bound over all $t \in [T]$ we get that with probability at least $1 - \frac{\delta}{2r}$,

$$\frac{1}{T}\sum_{t=1}^{T}\frac{1}{m}\sum_{j=1}^{m} w_{\boldsymbol{\alpha}_t}(h_t(x_j), y_j) \leq \frac{1}{T}\sum_{t=1}^{T} z_{\boldsymbol{\alpha}_t} + \frac{\epsilon}{10} \ , \tag{39}$$

which, recalling the definition of $w_{\boldsymbol{\alpha}}$, implies

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}_{i'\sim\boldsymbol{\alpha}_t}\Big[M(h_t,i')\Big] \le \frac{\epsilon}{10}\,. \tag{40}$$

Thus, we get that with probability $1-\frac{\delta}{2r}$,

$$\frac{1}{T}\sum_{t=1}^{T}M(h_t,i) \le \frac{\ln(r)}{T\eta}+\frac{\eta}{2}+\frac{\epsilon}{10} \le \frac{\epsilon}{2}\,. \tag{41}$$

By then also taking a union bound over $i=1,\dots,r$, we have with probability at least $1-\frac{\delta}{2}$ for all $i=1,\dots,r$ it holds that,

$$\frac{1}{T}\sum_{t=1}^{T}M(h_t,i) \le \frac{\epsilon}{2}\,. \tag{42}$$

Now consider the hypothesis $\widetilde{h}$ that, given any example x, returns $\frac{1}{T}\sum_{t=1}^{T}h_t(x)$. By Equation (42), with probability at least $1-\frac{\delta}{2}$,

$$\forall i=1,\dots,r,\;\; L_S^{w_i}(\widetilde{h}) \le z_i+\frac{\epsilon}{2}\,. \tag{43}$$

Finally, for each $t=1,\dots,T$ we apply standard compression-based generalization arguments to h_t (see, e.g., Shalev-Shwartz and Ben-David (2014, Theorem 30.2) and Alon et al. (2022, Lemma 42)). We obtain that, with probability at least $1-\frac{\delta}{2T}$ over the distribution of S, for every $t=1,\dots,T$:

$$L_{\mathcal{D}}^{w_i}(h_t) \le L_S^{w_i}(h_t)+O\left(\sqrt{\frac{\widehat{m}\log(mT/\delta)}{m}}\right). \tag{44}$$

Letting $m=\widetilde{O}\big(\frac{\widehat{m}}{\epsilon^2}\big)$, and by a union bound over $t\in[T]$, with probability at least $1-\delta$:

$$\forall i=1,\dots,r,\;\; L_{\mathcal{D}}^{w_i}(\widetilde{h}) \le z_i+\epsilon\,. \tag{45}$$

We conclude that the algorithm described above is a $(\boldsymbol{w},\boldsymbol{z})$-learner for $\mathcal{F}$.

B.2. Proof of Theorem 8

The following lemma is the key component used to prove Theorem 8.

Lemma 26 *Let $r\in\mathbb{N}$, let $\boldsymbol{w}=(w_1,\dots,w_r)$ where each $w_i:\mathcal{Y}^2\to[0,1]$ is a cost function, and let $\boldsymbol{z}\in[0,1]^r$. Fix a nonempty $J\subseteq\mathcal{Y}$ and $\boldsymbol{q}\in\Delta_J$, and assume that for every $\boldsymbol{\alpha}\in\Delta_r$ there exists $\boldsymbol{p}\in\Delta_{\mathcal{Y}}$ such that $\sum_{i=1}^{r}\alpha_i\cdot w_i(\boldsymbol{p},\boldsymbol{q})\le\langle\boldsymbol{\alpha},\boldsymbol{z}\rangle$. Then, there exists $\boldsymbol{p}^*\in\Delta_{\mathcal{Y}}$ such that for all $i=1,\dots,r$, it holds that $w_i(\boldsymbol{p}^*,\boldsymbol{q})\le z_i$.*

Proof Define the following zero-sum game. The set of pure strategies of the maximizing player is $[r]$, and the set of pure strategies of the minimizing player is $\mathcal{Y}$. The payoff matrix M is given as follows, with rows corresponding to labels $\ell\in\mathcal{Y}$ and columns to costs w_i, $i\in[r]$:

$$M(\ell,i)=\mathbb{E}_{j\sim\boldsymbol{q}}\,w_i(\ell,j)-z_i\,. \tag{46}$$

The expected payoff for mixed strategies $\boldsymbol{p}$ and $\boldsymbol{\alpha}$ is:

$$M(\boldsymbol{p}, \boldsymbol{\alpha}) \triangleq \mathbb{E}_{\ell \sim \boldsymbol{p}} \mathbb{E}_{i \sim \boldsymbol{\alpha}} M(\ell, i) = \sum_{i=1}^{r} \alpha_i w_i(\boldsymbol{p}, \boldsymbol{q}) - \langle \boldsymbol{\alpha}, \boldsymbol{z} \rangle \ . \tag{47}$$

Define:

$$\mathrm{V}(\boldsymbol{q}) \triangleq \max_{\boldsymbol{\alpha} \in \Delta_r} \min_{\boldsymbol{p} \in \Delta_{\mathcal{Y}}} M(\boldsymbol{p}, \boldsymbol{\alpha}) \ . \tag{48}$$

It remains to show that there exists $\boldsymbol{p}$ such that $w_i(\boldsymbol{p}, \boldsymbol{q}) \leq z_i$ for every $i = 1, \dots, r$. By assumption $\mathrm{V}(\boldsymbol{q}) \leq 0$, and so by von Neumann's Minimax Theorem (von Neumann, 1928):

$$\min_{\boldsymbol{p} \in \Delta_{\mathcal{Y}}} \max_{\boldsymbol{\alpha} \in \Delta_r} M(\boldsymbol{p}, \boldsymbol{\alpha}) \leq 0 \ . \tag{49}$$

In particular, there exists $\boldsymbol{p}^* \in \Delta_{\mathcal{Y}}$ such that for all $\boldsymbol{\alpha} \in \Delta_r$ we have $M(\boldsymbol{p}, \boldsymbol{\alpha}) \leq 0$, that is,

$$\sum_{i=1}^{r} \alpha_i w_i(\boldsymbol{p}^*, \boldsymbol{q}) \leq \langle \boldsymbol{\alpha}, \boldsymbol{z} \rangle \ . \tag{50}$$

Considering $\boldsymbol{\alpha} = \boldsymbol{e}_i$ (the i-th vector of the canonical basis of $\mathbb{R}^r$) for $i \in [r]$ shows that $w_i(\boldsymbol{p}^*, \boldsymbol{q}) \leq z_i$ for every $i \in [r]$, as claimed. ■

The next proposition has Theorem 8 as an immediate corollary.

Proposition 27 *Let $r \in \mathbb{N}$, and let $\boldsymbol{w} = (w_1, \dots, w_r)$ where each $w_i : \mathcal{Y}^2 \to [0, 1]$ is a cost function, $\boldsymbol{z} \in [0, 1]^r$, and $J \subseteq \mathcal{Y}$ nonempty. Then, the following are equivalent:*

1. *$(\boldsymbol{w}, \boldsymbol{z})$ is J-dice-attainable.*
2. *for every $\boldsymbol{\alpha} \in \Delta_r$ it holds that $\langle \boldsymbol{\alpha}, \boldsymbol{z} \rangle \geq \mathrm{V}_J(w_{\boldsymbol{\alpha}})$, where $w_{\boldsymbol{\alpha}} = \sum_{i=1}^r \alpha_i w_i$.*

Proof First, it is easy to see that $1 \implies 2$ holds, since by definition and by the assumption that $(\boldsymbol{w}, \boldsymbol{z})$ is J-dice-attainable, we get that for any $\boldsymbol{\alpha} \in \Delta_r$:

$$\max_{\boldsymbol{q} \in \Delta_J} \min_{\boldsymbol{p} \in \Delta_{\mathcal{Y}}} w_{\boldsymbol{\alpha}}(\boldsymbol{p}, \boldsymbol{q}) \leq \langle \boldsymbol{\alpha}, \boldsymbol{z} \rangle \ . \tag{51}$$

Then, by von Neumann's Minimax Theorem (von Neumann, 1928), we get that the claim holds.

Next, we prove $2 \implies 1$. By assumption, for every $\boldsymbol{\alpha} \in \Delta_r$ it holds that

$$\mathrm{V}_J(w_{\boldsymbol{\alpha}}) = \max_{\boldsymbol{q} \in \Delta_J} \min_{\boldsymbol{p} \in \Delta_{\mathcal{Y}}} \sum_{i=1}^{r} \alpha_i \cdot w_i(\boldsymbol{p}, \boldsymbol{q}) \leq \langle \boldsymbol{\alpha}, \boldsymbol{z} \rangle \ . \tag{52}$$

In particular, this implies that for any $\boldsymbol{q} \in \Delta_J$ and any $\boldsymbol{\alpha} \in \Delta_r$, there exists $\boldsymbol{p} \in \Delta_{\mathcal{Y}}$ such that

$$\sum_{i=1}^{r} \alpha_i \cdot w_i(\boldsymbol{p}, \boldsymbol{q}) \leq \langle \boldsymbol{\alpha}, \boldsymbol{z} \rangle \ .$$

Thus, by Lemma 26 we get that for any $\boldsymbol{q} \in \Delta_J$, there exists $\boldsymbol{p}^* \in \Delta_{\mathcal{Y}}$ such that for every $i \in [r]$,

$$w_i(\boldsymbol{p}^*, \boldsymbol{q}) \leq z_i \ .$$

As a consequence, we get that $(\boldsymbol{w}, \boldsymbol{z})$ is J-dice-attainable. ■

Appendix C. Multiclass Classification

In contrast to binary classification, boosting in the multiclass setting does not exhibit a clean dichotomy between "trivial" guarantees and "fully boostable" guarantees. A line of works (Mukherjee and Schapire, 2011; Brukhim et al., 2021, 2023b,a) has studied multiclass boosting when w is the standard 0-1 loss, and has shown a certain *multichotomy* determined by $k-1$ thresholds, $\frac{1}{2}, \frac{2}{3}, \ldots, \frac{k-1}{k}$, that replace the single boostability threshold $1/2$ of the binary case. For each such threshold, any learner achieving a loss below it can be "boosted" to a predictor that rules out *some* incorrect labels, albeit not all. Thus, for every example x the predictor returns a subset of candidate labels, also called a *list*. A learner that beats a threshold $\frac{\ell-1}{\ell}$ above in fact yields a list predictor with lists of size $\ell-1$, and it can be shown that the converse is also true. In this section we generalize this kind of results to the case of arbitrary cost functions, proving that these list boostability and list-versus-loss equivalences hold more broadly.

We start in Appendix C.1 by describing an algorithm that turns a (w, z)-learner into a list learner, proving the first part of Theorem C. Next, Section C.3 proves some "list boostability" results that focus on the list length, based on the framework of *List PAC learning* (Brukhim et al., 2022; Charikar and Pabbaraju, 2023). We turn to multi-objective losses in Section C.2, proving the first item of Theorem D. Finally, Appendix C.4 concludes by giving the lower bounds of both Theorem C and Theorem D.

C.1. Cost-sensitive boosting

As said, in multiclass classification one cannot in general boost a learner into one that predicts the correct label. However, as prior work has shown, one can boost a learner into a *list learner*. This is a learner whose output hypothesis maps each $x \in \mathcal{X}$ to a set of labels, rather than to one single label. Formally, a *list function* is any map $\mu : \mathcal{X} \to 2^{\mathcal{Y}}$. A list learner is then a learner that outputs a list function μ. The performance of such a list function μ is typically measured in two ways. First, one wants $\mu(x)$ to contain the true label $f(x)$ with good probability. Second, one wants $\mu(x)$ to somehow be "close" to $f(x)$; for instance, in the case of standard 0-1 cost, this is measured by the length $|\mu(x)|$ of $\mu(x)$. All these guarantees are meant, as usual, over the target distribution $\mathcal{D}$. One can then ask the following question: given a (w, z)-learner, how good a list function μ can one construct? In order to answer this question in the cost-sensitive setting, we need to generalize list length to a more nuanced measure—one that is based once again on the values of games.

Definition 28 ((w, z)-boundedness) *Let $w : \mathcal{Y}^2 \to [0, 1]$ be a cost function and let $z \geq 0$. A list function $\mu : \mathcal{X} \to 2^{\mathcal{Y}}$ is* (w, z)-bounded *if, for every $x \in \mathcal{X}$, the list $\mu(x)$ satisfies $\mathrm{V}_{\mu(x)}(w) \leq z$.*

Definition 29 ((w, z)-list learner) *Let $w : \mathcal{Y}^2 \to [0, 1]$ be a cost function and let $z \geq 0$. An algorithm $\mathcal{L}$ is a* (w, z)-list learner *for a class $\mathcal{F} \subseteq \mathcal{Y}^{\mathcal{X}}$ if there is a function $m_{\mathcal{L}} : (0, 1)^2 \to \mathbb{N}$ such that for every $f \in \mathcal{F}$, every distribution $\mathcal{D}$ over $\mathcal{X}$, and every $\epsilon, \delta \in (0, 1)$ the following claim holds. If S is a sample of $m_{\mathcal{L}}(\epsilon, \delta)$ examples drawn i.i.d. from $\mathcal{D}$ and labeled by f, then $\mathcal{L}(S)$ returns a (w, z)-bounded list function μ such that $\mathbb{P}_{x\sim\mathcal{D}}[f(x) \notin \mu(x)] \leq \epsilon$ with probability at least $1 - \delta$.*

The definitions above mirror closely the standard ones used in list learning, where the quality of a list $\mu(x)$ is measured by its length $|\mu(x)|$. The crucial difference is that, now, we are instead using the value of the restricted game $\mathrm{V}_{\mu(x)}(w)$. To get some intuition why this is the right choice, consider again the case where w is the standard 0-1 cost. In that case it is well known that $V_J(w) = 1 - 1/|J|$

for every (nonempty) $J \subseteq \mathcal{Y}$. Then, by Definition 28 and Definition 29 a (w, z)-list learner is one that outputs lists of length at most $\ell = \min(k, \lfloor \frac{1}{1-z} \rfloor)$. That is, for the 0-1 cost Definition 29 yields the standard notion of ℓ-list learner used in prior work on multiclass boosting. Now, for the special case of the 0-1 cost, Brukhim et al. (2023a) showed that any (w, z)-learner is equivalent to an ℓ-list learner, with ℓ as defined above, in the sense that each one of them can be converted into the other. We show that this remains true for *every* cost function w, if one replaces $|\mu(x)|$ with $\mathrm{V}_{\mu(x)}(w)$, and therefore the notion of ℓ-list learner with the notion of (w, z)-list learner. That is, we prove that (w, z)-learners and (w, z)-list learners are equivalent. Formally, we prove:

Theorem 30 *Let $\mathcal{Y} = [k]$, let $w : \mathcal{Y}^2 \to [0, 1]$ be a cost function, let $\mathcal{F} \subseteq \mathcal{Y}^{\mathcal{X}}$, and let $z \geq 0$. Then:*

- *Every (w, z)-learner for $\mathcal{F}$ can be converted into a (w, z)-list learner for $\mathcal{F}$.*
- *Every (w, z)-list learner for $\mathcal{F}$ can be converted into a (w, z)-learner for $\mathcal{F}$.*

Theorem 30 is the main technical result of the present sub-section, and implies easily the first item of Theorem C. And again, the above-mentioned result of Brukhim et al. (2023a) is a special case of Theorem 30, obtained for w being the standard 0-1 cost. Moreover, one can easily show that all these results do not hold if one uses $|\mu(x)|$ in place of $\mathrm{V}_{\mu(x)}(w)$. Our game-theoretical perspective therefore seems the right point of view for characterizing multiclass boostability. Let us see how the first item of Theorem C follows from Theorem 30. Recall the thresholds defined in Equation (6):

$$0 = v_1(w) < \cdots < v_\tau(w) = V(w), \tag{53}$$

where the first equalities hold because $v_J(w) = 0$ for any singleton J and, by convention, for the empty set. Given $z \geq 0$, let $n = n(z)$ be the largest integer such that $v_n(w) \leq z$. Now, suppose we have a (w, z)-learner $\mathcal{A}$ for some $\mathcal{F}$. By the first item of Theorem 30, there exists a (w, z)-list learner $\mathcal{L}$ for $\mathcal{F}$, too. However, by definition of n every $J \subseteq \mathcal{Y}$ with $\mathrm{V}_J(w) \leq z$ satisfies $\mathrm{V}_J(w) \leq v_n$. Therefore, $\mathcal{L}$ actually returns list functions that are $(w, v_n + \sigma)$-bounded, for $\sigma > 0$ arbitrarily small. That is, $\mathcal{L}$ is actually a (w, v_n)-list learner for $\mathcal{F}$. By the second item of Theorem 30, then, there exists a (w, v_n)-learner for $\mathcal{F}$.

The rest of the subsection is therefore devoted to prove Theorem 30: the first item in Appendix C.1.1, and the second one in Appendix C.1.2.

C.1.1. TURNING A (w, z)-LEARNER INTO A (w, z)-LIST LEARNER

We prove the first item of Theorem 30 by giving an algorithm, Algorithm 3. The algorithm is very similar to the one for the binary case. And, like for that case, we define a certain *margin* that the learner has, in order to bound the number of "boosting" rounds of the algorithm. For the multiclass case, however, the definition is in general slightly different.

Definition 31 (Margin) *Let $w : \mathcal{Y}^2 \to [0, 1]$ be a cost function and let $z \geq 0$. The margin of a (w, z)-learner is $\gamma = \min\{\mathrm{V}_J(w) - z : J \subseteq \mathcal{Y}, \mathrm{V}_J(w) > z\}$, or $\gamma = 0$ whenever the set is empty.*

Note that $\gamma = 0$ only when $z \geq \mathrm{V}(w)$. In that case, the first item of Theorem 30 holds trivially by just considering the trivial list learner that outputs the constant list function $\mu(x) = \mathcal{Y}$. Hence, in what follows we consider always the case $\gamma > 0$. We shall prove:

Algorithm 3 Boosting a (w, z)-learner to a $(w, z + \sigma)$-list learner

Input: Sample $S = (x_i, y_i)_{i=1}^m$; (w, z)-learner $\mathcal{A}$; parameters $T, \eta, \widehat{m}, \sigma$

```
1: Initialize: $D_1(i) = 1$ for all $i = 1, ..., m$.
2: for $t = 1, \dots, T$ do
3:     Compute the distribution $\mathcal{D}_t \triangleq \frac{D_t}{\sum_i D_t(i)}$ over $S$
4:     Draw a set $S_t$ of $\widehat{m}$ labeled examples i.i.d. from $\mathcal{D}_t$ and obtain $h_t = \mathcal{A}(S_t)$.
5:     For every $i = 1, \dots, m$ let:
           $$D_{t+1}(i) \triangleq D_t(i) \cdot e^{\eta \cdot w(h_t(x_i), y_i)} .$$
6: end for
7: For all $x \in \mathcal{X}$ let $\boldsymbol{p}^x \triangleq \frac{1}{T} \sum_{t=1}^T h_t(x) \in \Delta_{\mathcal{Y}}$.
8: return $\mu_S : \mathcal{X} \to 2^{\mathcal{Y}}$ such that, for all $x \in \mathcal{X}$,
           $$\mu_S(x) \triangleq \left\{ y \in \mathcal{Y} : w(\boldsymbol{p}^x, y) \le z + \sigma \right\} .$$
```

Theorem 32 (Weak $\Rightarrow$ List Learning) *Let $\mathcal{F} \subseteq \mathcal{Y}^{\mathcal{X}}$ and let $w : \mathcal{Y}^2 \to [0, 1]$ be a cost function. If $\mathcal{A}$ is a (w, z)-learner for $\mathcal{F}$ with margin $\gamma > 0$, then Algorithm 3, with the choice of parameters of Lemma 33 and $\sigma = \frac{2}{3}\gamma$, is a (w, z)-list learner for $\mathcal{F}$. Under the same assumptions, Algorithm 3 makes $T = O\big(\frac{\ln(m)}{\gamma^2}\big)$ oracle calls to $\mathcal{A}$, where m is the size of the input sample, and has sample complexity $m(\epsilon, \delta) = \widetilde{O}\Big(m_0\big(\frac{\gamma}{3}, \frac{\delta}{2T}\big) \cdot \frac{\ln(1/\delta)}{\epsilon \cdot \gamma^2}\Big)$.*

Theorem 32 follows immediately from two technical results stated below. The first result, Lemma 33, proves that Algorithm 3 can turn a (w, z)-learner $\mathcal{A}$ into a list function μ_S that is $(w, z+\sigma)$-bounded and that with probability $1 - \delta$ is consistent with S (i.e., $y_i \in \mu(x_i)$ for every $(x_i, y_i) \in S$), for $\sigma, \delta > 0$ as small as desired. The second result, Lemma 34, shows that the list function μ has small generalization error: if the input sample S of Algorithm 3 is sufficiently large, then $\mathbb{P}_{x \sim \mathcal{D}}[f(x) \in \mu(x)] \ge 1 - \epsilon$, where $\epsilon > 0$ can be chosen as small as desired.

Lemma 33 *Let $\mathcal{Y} = [k]$, let $\mathcal{F} \subseteq \mathcal{Y}^{\mathcal{X}}$, and let $\mathcal{A}$ be a (w, z)-learner for $\mathcal{F}$ with sample complexity m_0. Fix any $f \in \mathcal{F}$, any distribution $\mathcal{D}$ over $\mathcal{X}$, and any $\sigma, \delta \in (0, 1)$. Then, given a multiset $S = \{(x_1, y_1), \dots, (x_m, y_m)\}$ of m examples formed by i.i.d. points $x_1, \dots, x_m \sim \mathcal{D}$ labeled by f, oracle access to $\mathcal{A}$, and parameters $T = \left\lceil \frac{8\ln(m)}{\sigma^2} \right\rceil$, $\eta = \sqrt{\frac{2\ln(m)}{T}}$, and $\widehat{m} = m_0\big(\frac{\sigma}{2}, \frac{\delta}{T}\big)$, Algorithm 3 returns a list function μ_S that is $(w, z + \sigma)$-bounded and that, with probability at least $1 - \delta$, satisfies:*

$$y_i \in \mu_S(x_i) \qquad \forall i = 1, \dots, m .$$

Proof First, we prove that μ_S is $(w, z+\sigma)$-bounded. Fix any $x \in \mathcal{X}$. By construction, Algorithm 3 lets $y \in \mu_S$ if and only if $w(\boldsymbol{p}^x, y) \le z + \sigma$. Then:

$$\mathrm{V}_{\mu_S(x)}(w) = \min_{\boldsymbol{p} \in \Delta_{\mathcal{Y}}} \max_{\boldsymbol{q} \in \Delta_{\mu_S(x)}} w(\boldsymbol{p}, \boldsymbol{q}) \le \max_{\boldsymbol{q} \in \Delta_{\mu_S(x)}} w(\boldsymbol{p}^x, \boldsymbol{q}) = \max_{y \in \mu_S(x)} w(\boldsymbol{p}^x, y) \le z + \sigma . \tag{54}$$

We now turn to proving that with probability at least $1-\delta$ we have $y_i \in \mu_S(x_i)$ for all $i \in [m]$. The proof is similar to that of Lemma 13. For any given $i \in [m]$, the analysis of Hedge and the definitions of η and T yield:

$$\frac{1}{T}\sum_{t=1}^{T} w(h_t(x_i), y_i) \le \frac{\sigma}{2} + \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}_{j\sim\mathcal{D}_t}\Big[w(h_t(x_j), y_j)\Big] . \tag{55}$$

Furthermore, since $\mathcal{A}$ is a (w,z)-learner for $\mathcal{F}$, and given the choices of $\widehat{m}$ and $h_t = \mathcal{A}(S_t)$, by a union bound over $t \in [T]$ we have that with probability at least $1-\delta$:

$$\frac{1}{T}\sum_{t=1}^{T} \mathbb{E}_{j\sim\mathcal{D}_t}\Big[w(h_t(x_j), y_j)\Big] \le z + \frac{\sigma}{2} . \tag{56}$$

Conditioning on Equation (56), we prove that $y_i \in \mu_S(x_i)$ for every $i \in [m]$. By Equations (55) and (56), by definition of $\boldsymbol{p}^x \in \Delta_{\mathcal{Y}}$ from Algorithm 3, and by linearity of w, for every $i \in [m]$ we have

$$w(\boldsymbol{p}^x, y_i) = \frac{1}{T}\sum_{t=1}^{T} w(h_t(x_i), y_i) \le z + \sigma . \tag{57}$$

which in turn implies that $y_i \in \mu_S(x_i)$ by construction of μ_S. This concludes the proof. ■

In a similar way as Lemma 14 does for the binary case, we show that the list function μ_S returned by Algorithm 3 has small generalization error provided that the sample S is sufficiently large. The main idea to prove generalization is again via a sample compression scheme, but relying instead on a novel result by Brukhim et al. (2023a) for multiclass classification; note that, while their result from Brukhim et al. (2023a) is stated in terms of list functions whose outputs are s-uples in $\mathcal{Y}^s$ for some $s < k$, their same proof applies to the (w,z)-bounded list functions output by our Algorithm 3.

Lemma 34 *Assume the setting of Lemma 33. For any $\epsilon, \delta \in (0,1)$, if the size m of the sample given to Algorithm 3 satisfies*

$$m \ge \frac{\ln(2/\delta)}{\epsilon} + m_0\left(\frac{\sigma}{2}, \frac{\delta}{2T}\right) \cdot T \cdot \left(1 + \frac{\ln(m)}{\epsilon}\right) ,$$

then the output μ_S of Algorithm 3 satisfies $\mathbb{P}_{x\sim\mathcal{D}}[f(x) \notin \mu_S(x)] \le \epsilon$ with probability at least $1-\delta$. Therefore, Algorithm 3 is a $(w, z+\sigma)$-list learner for $\mathcal{F}$ with sample complexity $m(\epsilon,\delta) = \widetilde{O}\Big(m_0\big(\frac{\sigma}{2}, \frac{\delta}{2T}\big) \cdot \frac{\ln(1/\delta)}{\epsilon\cdot\sigma^2}\Big)$.

Proof By analogy with the proof of Lemma 14, we first apply Lemma 33 with $\delta/2$ in place of δ in order to obtain an $(w, z+\sigma)$-bounded list function μ_S consistent with S with probability at least $1-\delta/2$. We then apply a compression-based generalization argument. To do so, we remark once again that one can construct a compression scheme for μ_S of size equal to the total size of the samples on which $\mathcal{A}$ operates, which is equal to $\kappa = \widehat{m} \cdot T$. The main difference with the binary case is that we rely on the generalization bound for a sample compression scheme for lists as per

Theorem 6 of Brukhim et al. (2023a), with $\delta/2$ in place of δ; we can employ this generalization bound thanks to the consistency of μ_S with S (with sufficiently large probability). This implies

$$\mathbb{P}_{x\sim\mathcal{D}}\big[f(x)\notin\mu_S(x)\big] \le \frac{\kappa\ln(m)+\ln(2/\delta)}{m-\kappa} \tag{58}$$

with probability at least $1-\delta/2$. By similar calculations as in Equations (18) and (19), and replacing the values of κ and $\widehat{m}$, the right-hand side of the above inequality can be easily shown to be at most ϵ whenever

$$m \ge \frac{\ln(2/\delta)}{\epsilon} + m_0\left(\frac{\sigma}{2},\frac{\delta}{2T}\right)\cdot T\cdot\left(1+\frac{\ln(m)}{\epsilon}\right) \ . \tag{59}$$

A union bound concludes the proof for the first part of the claim, since it shows that Algorithm 3 is an $(w, z+\sigma)$-list learner for $\mathcal{F}$. The claim on the sample complexity of Algorithm 3 then follows by straightforward calculations and the definition of $T = O\big(\frac{\ln(m)}{\sigma^2}\big)$. ■

At this point, we have all the ingredients and tools to prove that we can construct a (w,z)-list learner from a (w,z)-learner.

Proof of Theorem 32. Consider Algorithm 3 under the same assumptions of Lemma 33, and set $\sigma=\frac{2}{3}\gamma$ as per assumption. By Lemma 33 and Lemma 34, we know that Algorithm 3 is a $(w, z+\frac{2}{3}\gamma)$-list learner with sample complexity $m(\epsilon,\delta)=\widetilde{O}\big(m_0\big(\frac{\gamma}{3},\frac{\delta}{2T}\big)\cdot\frac{\ln(1/\delta)}{\epsilon\cdot\gamma^2}\big)$ that performs $O\big(\frac{\ln(m)}{\gamma^2}\big)$ oracle calls to the (w,z)-learner $\mathcal{A}$. Moreover, we can immediately conclude that Algorithm 3 is also a (w,z)-list learner because $z\ge v_n$ and $z+\frac{2}{3}\gamma < z+\gamma = v_{n+1}$, by definitions of $n=n(z)$ and γ. ■

C.1.2. TURNING A (w,z)-LIST LEARNER INTO A (w,z)-LEARNER

We prove that every (w,z)-list learner can be converted into a (w,z)-learner.

Lemma 35 (List ⇒ Weak Learning) *There exists an algorithm $\mathcal{B}$ that for every $\mathcal{F}\subseteq\mathcal{Y}^{\mathcal{X}}$ satisfies what follows. Let $w:\mathcal{Y}^2\to[0,1]$ be a cost function and let $z\ge 0$. Given oracle access to a (w,z)-list learner $\mathcal{L}$ for $\mathcal{F}$ with sample complexity $m_{\mathcal{L}}$, algorithm $\mathcal{B}$ is a (w,z)-learner for $\mathcal{F}$ with sample complexity $m_{\mathcal{L}}$ that makes one oracle call to $\mathcal{L}$.*

Proof Fix any $f\in\mathcal{F}$ and any distribution $\mathcal{D}$ over $\mathcal{X}$, and suppose $\mathcal{B}$ is given a sample S of size $m\ge m_{\mathcal{L}}(\epsilon,\delta)$ consisting of examples drawn i.i.d. from $\mathcal{D}$ and labeled by f. First, $\mathcal{B}$ calls $\mathcal{L}$ on S. By hypothesis, then, $\mathcal{L}$ returns a (w,z)-bounded $\mu_S:\mathcal{X}\to\mathcal{Y}^{\mathcal{X}}$ that with probability at least $1-\delta$ satisfies:

$$\mathbb{P}_{x\sim\mathcal{D}}\big[f(x)\notin\mu_S(x)\big]\le\epsilon \ . \tag{60}$$

Conditioning on the event above from this point onwards, we give a randomized predictor h_S such that $L^w_{\mathcal{D}}(h_S)\le z+\epsilon$. First, for any nonempty $J\subseteq\mathcal{Y}$, let $\boldsymbol{p}_J\in\Delta_{\mathcal{Y}}$ be the minimax distribution achieving the value of the game restricted to J, i.e.,

$$\boldsymbol{p}_J = \arg\min_{\boldsymbol{p}\in\Delta_{\mathcal{Y}}}\left(\max_{\boldsymbol{q}\in\Delta_J} w(\boldsymbol{p},\boldsymbol{q})\right) \ . \tag{61}$$

Note that $\boldsymbol{p}_J$ can be efficiently computed via a linear program. Then, simply define $h_S(x) \sim \boldsymbol{p}_{\mu_S(x)}$. Let us analyze the loss of h_S under $\mathcal{D}$. First, by the law of total expectation, and since $\|w\|_\infty \leq 1$,

$$L^w_{\mathcal{D}}(h_S) \leq \mathbb{P}_{x\sim D}[f(x) \notin \mu_S(x)] + \sum_J \mathbb{P}_{x\sim\mathcal{D}}[\mu_S(x) = J \wedge f(x) \in J] \cdot L^w_{\mathcal{D}_J}(h_S) \ , \tag{62}$$

where the summation is over all $J \subseteq \mathcal{Y}$ with $\mathbb{P}_{x\sim\mathcal{D}}[\mu_S(x) = J] > 0$, and $\mathcal{D}_J$ is the distribution obtained from $\mathcal{D}$ by conditioning on the event $\mu_S(x) = J \wedge f(x) \in J$. Consider the right-hand side of Equation (62). By Equation (60), the first term is at most ϵ. For the second term, denote by $\boldsymbol{q}_J$ the marginal of $\mathcal{D}_J$ over $\mathcal{Y}$; note that, crucially, $\boldsymbol{q}_J \in \Delta_J$. Therefore, by definition of h_S:

$$L^w_{\mathcal{D}_J}(h_S) = w(\boldsymbol{p}_J, \boldsymbol{q}_J) \leq \max_{\boldsymbol{q}\in\Delta_J} w(\boldsymbol{p}_J, \boldsymbol{q}) = \mathrm{V}_J(w) \leq z \ , \tag{63}$$

where the last inequality holds as $J = \mu_S(x)$ and μ_S is (w, z)-bounded. Using Equations (60) and (63) in Equation (62) shows that $L^w_{\mathcal{D}}(h_S) \leq z + \epsilon$. This concludes the proof. ∎

C.2. Multi-objective losses

In this subsection we prove the first item of Theorem D. Coupled with Lemma 47 below, this proves Theorem D. Both results require the following definition. Let $\mathrm{av}(\boldsymbol{w}, \boldsymbol{z})$ denote the family of all subsets of $\mathcal{Y}$ which are *avoided* by $(\boldsymbol{w}, \boldsymbol{z})$, i.e.,

$$\mathrm{av}(\boldsymbol{w}, \boldsymbol{z}) \triangleq \{J \subseteq \mathcal{Y} : \boldsymbol{z} \notin D_J(\boldsymbol{w})\}.$$

The algorithm proposed below then scales with $N = |\,\mathrm{av}(\boldsymbol{w}, \boldsymbol{z})|$, which may be exponential in $|\mathcal{Y}|$. We remark that it suffices to consider only the *minimally-sized* avoided lists in $\mathrm{av}(\boldsymbol{w}, \boldsymbol{z})$, i.e., if $J \subseteq J'$ and $\boldsymbol{z} \notin D_J(\boldsymbol{w})$ then $J' \notin \mathrm{av}(\boldsymbol{w}, \boldsymbol{z})$. However, in the worst case N remains of the same order and so we keep the definition as given above for simplicity. Lemma 47 will then give a lower bound showing that this condition is, in some sense, necessary.

Theorem 36 *Let $\mathcal{Y} = [k]$, let $\boldsymbol{w} = (w_1, \ldots, w_r)$ where each $w_i : \mathcal{Y}^2 \to [0, 1]$ is a cost function, and let $\boldsymbol{z} \in [0, 1]^r$. Let $\boldsymbol{z}' \in [0, 1]^r$ such that $\boldsymbol{z} \preceq_{\boldsymbol{w}} \boldsymbol{z}'$. Assume there exists a $(\boldsymbol{w}, \boldsymbol{z})$-learner for a class $\mathcal{F} \subseteq \mathcal{Y}^{\mathcal{X}}$. Then there exists a $(\boldsymbol{w}, \boldsymbol{z}')$-learner for $\mathcal{F}$.*

Proof By assumption, for every $J \subseteq \mathcal{Y}$ such that $J \in \mathrm{av}(\boldsymbol{w}, \boldsymbol{z}')$ it holds that $J \in \mathrm{av}(\boldsymbol{w}, \boldsymbol{z})$. Let $\boldsymbol{\alpha} \in \Delta_r$, and denote $w_{\boldsymbol{\alpha}} = \boldsymbol{\alpha} \cdot \boldsymbol{w}$ and $z'_{\boldsymbol{\alpha}} = \boldsymbol{\alpha} \cdot \boldsymbol{z}'$. Then, by Lemma 37 we get that the $(\boldsymbol{w}, \boldsymbol{z})$-learner can be used to obtain a $(w_{\boldsymbol{\alpha}}, z'_{\boldsymbol{\alpha}})$-learner for $\mathcal{F}$. Since this holds for every $\boldsymbol{\alpha} \in \Delta_r$, then by Theorem 7 this implies the existence of a $(\boldsymbol{w}, \boldsymbol{z}')$-learner as well, which completes the proof. ∎

Lemma 37 *Let $\boldsymbol{w} = (w_1, \ldots, w_r)$ where each $w_i : \mathcal{Y}^2 \to [0, 1]$ is a cost function, and let $\boldsymbol{z}, \boldsymbol{z}' \in [0, 1]^r$ such that $\boldsymbol{z} \preceq_{\boldsymbol{w}} \boldsymbol{z}'$. Assume there exists a $(\boldsymbol{w}, \boldsymbol{z})$-learner $\mathcal{A}$ for a class $\mathcal{F} \subseteq \mathcal{Y}^{\mathcal{X}}$. Then for every $\boldsymbol{\alpha} \in \Delta_r$ there exists a $(w_{\boldsymbol{\alpha}}, z'_{\boldsymbol{\alpha}})$-learner for $\mathcal{F}$.*

Proof First, by Lemma 39 we get that $\mathcal{A}$ can be used to obtain an algorithm $\mathcal{L}$ that is a $\mathrm{av}(\boldsymbol{w}, \boldsymbol{z}')$-list learner for $\mathcal{F}$. Next, we show that $\mathcal{L}$ can be boosted to a $(w_{\boldsymbol{\alpha}}, z'_{\boldsymbol{\alpha}})$-learner. Fix any $\boldsymbol{\alpha} \in \Delta_r$. First, we argue that $\mathcal{L}$ is in fact a $(w_{\boldsymbol{\alpha}}, z'_{\boldsymbol{\alpha}})$-list learner (see Definition 29). This holds since for every

$J \subseteq \mathcal{Y}$ such that $z'_{\boldsymbol{\alpha}} < V_J(w_{\boldsymbol{\alpha}})$, it must be that $J \in \operatorname{av}(\boldsymbol{w}, \boldsymbol{z}')$, by definition and by Proposition 27. In particular, for any μ list function outputted by $\mathcal{L}$, and for every $x \in \mathcal{X}$ it holds that $J \not\subseteq \mu(x)$ and so $V_{\mu(x)}(w_{\boldsymbol{\alpha}}) \leq z'_{\boldsymbol{\alpha}}$. We conclude that $\mathcal{L}$ is a $(w_{\boldsymbol{\alpha}}, z'_{\boldsymbol{\alpha}})$-list learner, which by Lemma 35 implies the existence of a $(w_{\boldsymbol{\alpha}}, z'_{\boldsymbol{\alpha}})$-learner. ■

Definition 38 ($\operatorname{av}(\boldsymbol{w}, \boldsymbol{z})$-list learner) *An algorithm $\mathcal{L}$ is a* $\operatorname{av}(\boldsymbol{w}, \boldsymbol{z})$-list learner *for a class $\mathcal{F} \subseteq \mathcal{Y}^{\mathcal{X}}$ if there is a function $m_{\mathcal{L}} : (0,1)^2 \to \mathbb{N}$ such that for every $f \in \mathcal{F}$, every distribution $\mathcal{D}$ over $\mathcal{X}$, and every $\epsilon, \delta \in (0,1)$ the following claim holds. If S is a sample of $m_{\mathcal{L}}(\epsilon, \delta)$ examples drawn i.i.d. from $\mathcal{D}$ and labeled by f, then $\mathcal{L}(S)$ returns a list function $\mu : \mathcal{X} \to 2^{\mathcal{Y}}$ such that $\mathbb{P}_{x\sim\mathcal{D}}\left[f(x) \notin \mu(x)\right] \leq \epsilon$ with probability $1-\delta$, and such that for every $x \in \mathcal{X}$ and every $J \in \operatorname{av}(\boldsymbol{w}, \boldsymbol{z})$ it holds that $J \not\subseteq \mu(x)$.*

Lemma 39 *Let $\mathcal{Y} = [k]$, let $\boldsymbol{w} = (w_1, \ldots, w_r)$ where each $w_i : \mathcal{Y}^2 \to [0,1]$ is a cost function, and let $\boldsymbol{z}, \boldsymbol{z}' \in [0,1]^r$ such that $\boldsymbol{z} \preceq_{\boldsymbol{w}} \boldsymbol{z}'$. Let $\mathcal{F} \subseteq \mathcal{Y}^{\mathcal{X}}$ and assume there exists a $(\boldsymbol{w}, \boldsymbol{z})$-learner for $\mathcal{F}$. Then, there exists an $\operatorname{av}(\boldsymbol{w}, \boldsymbol{z}')$-list learner for $\mathcal{F}$.*

Proof Let $\mathcal{A}$ be any $(\boldsymbol{w}, \boldsymbol{z})$-learner for $\mathcal{F}$. The idea is to run Algorithm 3 with $\mathcal{A}$ in the role of a $(w_{\boldsymbol{\alpha}}, z_{\boldsymbol{\alpha}})$ for several different $\boldsymbol{\alpha}$, so to obtain a set of list learners each of which avoids a specific $J \in \operatorname{av}(\boldsymbol{w}, \boldsymbol{z}')$. We then prove that aggregating those learners via list intersection yields the sought $\operatorname{av}(\boldsymbol{w}, \boldsymbol{z}')$-list learner.

First, fix any $J \in \operatorname{av}(\boldsymbol{w}, \boldsymbol{z}')$. As $\boldsymbol{z} \preceq_{\boldsymbol{w}} \boldsymbol{z}'$, then $J \in \operatorname{av}(\boldsymbol{w}, \boldsymbol{z})$, hence $(\boldsymbol{w}, \boldsymbol{z})$ is not J-dice-attainable. Then, by Proposition 27 there exists $\boldsymbol{\alpha} \in \Delta_r$ such that

$$z_{\boldsymbol{\alpha}} \coloneqq \langle \boldsymbol{\alpha}, \boldsymbol{z} \rangle < \mathrm{V}_J(w_{\boldsymbol{\alpha}}), \tag{64}$$

where $w_{\boldsymbol{\alpha}} = \sum_{i=1}^{r} \alpha_i w_i$. Thus, $\mathcal{A}$ is also a $(w_{\boldsymbol{\alpha}}, z_{\boldsymbol{\alpha}})$-learner for $\mathcal{F}$, and $z_{\boldsymbol{\alpha}} < \mathrm{V}_J(w_{\boldsymbol{\alpha}})$. Observe that the above holds for each $J \in \operatorname{av}(\boldsymbol{w}, \boldsymbol{z}')$. That is, for each such J there exists $\boldsymbol{\alpha}_J \in \Delta_r$ for which $\mathcal{A}$ is also a $(w_{\boldsymbol{\alpha}_J}, z_{\boldsymbol{\alpha}_J})$-learner for $\mathcal{F}$, and $z_{\boldsymbol{\alpha}_J} < \mathrm{V}_J(w_{\boldsymbol{\alpha}_J})$.

Next, we describe the $\operatorname{av}(\boldsymbol{w}, \boldsymbol{z}')$-list learning algorithm $\mathcal{L}$. Fix any $f \in \mathcal{F}$, any distribution $\mathcal{D}$ over $\mathcal{X}$, and any $\delta, \epsilon > 0$. Let $S = (x_i, y_i)_{i=1}^{m}$ be a set of m examples drawn independently from $\mathcal{D}$ and labeled by f, for m to be defined later.

Consider then running Algorithm 3 using $\mathcal{A}$ as a $(w_{\boldsymbol{\alpha}_J}, z_{\boldsymbol{\alpha}_J})$-learner. As for the parameters, use $\sigma \coloneqq \sigma_J = \frac{\gamma_J}{2}$, where γ_J is the margin of $\mathcal{A}$ (see Definition 31), and $\widehat{m} = m_0(\sigma, \frac{\delta'}{2N})$, where $N = |\operatorname{av}(\boldsymbol{w}, \boldsymbol{z}')|$ and m_0 is the sample complexity of $\mathcal{A}$. Set the remaining parameters, T and η, as in Lemma 33. Let μ_J be the resulting list function.

For each such run of the algorithm—for each $J \in \operatorname{av}(\boldsymbol{w}, \boldsymbol{z}')$—we obtain a different list function μ_J. The aggregate list learner $\mathcal{L}$ returns the aggregate list function μ defined by:

$$\mu(x) = \bigcap_{J \in \operatorname{av}(\boldsymbol{w}, \boldsymbol{z}')} \mu_J(x), \qquad \forall x \in \mathcal{X}\,. \tag{65}$$

We shall now prove that $\mathcal{L}$ is indeed a $\operatorname{av}(\boldsymbol{w}, \boldsymbol{z}')$-list learner.

First, by Lemma 33, for every J the list function μ_J is $(w_{\boldsymbol{\alpha}_J}, z_{\boldsymbol{\alpha}_J} + \sigma_J)$-bounded. That is, for every $x \in \mathcal{X}$ it holds that $\mathrm{V}_{\mu_J(x)}(w_{\boldsymbol{\alpha}_J}) \leq z_{\boldsymbol{\alpha}_J} + \sigma_J = z_{\boldsymbol{\alpha}_J} + \frac{2}{3}\gamma_J$. Then, it is easy to see that this implies $J \not\subseteq \mu_J(x)$ for all $x \in \mathcal{X}$. Indeed, by the choice of σ_J above, and by definition of margin, we have

$$\mathrm{V}_{\mu_J(x)}(w_{\boldsymbol{\alpha}_J}) \leq z_{\boldsymbol{\alpha}_J} + \frac{\gamma_J}{2} < \mathrm{V}_J(w_{\boldsymbol{\alpha}_J}), \tag{66}$$

which implies $J \not\subseteq \mu_J(x)$ as claimed. Thus, in particular, we also get that the aggregate list function μ satisfies that for every $x \in \mathcal{X}$ it holds that $J \not\subseteq \mu(x)$.

Second, the list function μ satisfies $\mathbb{P}_{x\sim D}[f(x) \notin \mu(x)] \leq \epsilon$ with probability at least $1 - \frac{\delta}{2}$. Indeed, by Lemma 33, with probability at least $1 - \frac{\delta}{2N}$ the list μ_J satisfies $y_i \in \mu_J(x_i)$ for all $i \in [m]$. By a union bound, then, $y_i \in \mu(x_i)$ for all $i \in [m]$ with probability at least $1 - \frac{\delta}{2}$. By the same compression-based generalization analysis of Lemma 34, it follows that $\mathbb{P}_{x\sim\mathcal{D}}[f(x) \notin \mu(x)] \leq \epsilon$ with probability at least $1 - \delta$, concluding the proof that $\mathcal{L}$ is a $\mathrm{av}(\boldsymbol{w}, \boldsymbol{z}')$-list learning algorithm. Using the choice of parameters above, the sample complexity can be easily shown to be $m(\epsilon, \delta) = \widetilde{O}\Big(N \cdot m_0\big(\gamma^*, \frac{\delta}{NT}\big) \cdot \frac{\ln(N/\delta)}{\epsilon\cdot\gamma^{*2}}\Big)$, where $\gamma^* = \min_J \gamma_J$. ■

C.3. Multiclass boosting via s-list PAC learning

In this section we aim to convert weak learners to list learners, with a fixed list size. This is, in some sense, a coarsening of the previous subsection, which also subsumes previous work by Brukhim et al. (2023a), where the authors demonstrate $k - 1$ thresholds which determine the achievable list sizes. To that end, for every $s = 2, \ldots, k$ define the following coarsening of s-the critical thresholds of w:

$$v_{\underline{s}}(w) \triangleq \min\left\{\mathrm{V}_J(w) : J \in \binom{[k]}{s}\right\}, \qquad v_{\overline{s}}(w) \triangleq \max\left\{\mathrm{V}_J(w) : J \in \binom{[k]}{s}\right\}. \tag{67}$$

Clearly, for all $s \in \{2, \ldots, k\}$ we have that $v_{\underline{s}}(w) \leq v_{\overline{s}}(w)$. It is also easy to see that $0 \leq v_{\underline{2}}(w) \leq \ldots \leq v_{\underline{k}}(w)$, and $0 \leq v_{\overline{2}}(w) \leq \ldots \leq v_{\overline{k}}(w)$. When clear from context, we omit w and denote the thresholds by $v_{\underline{s}}$ and $v_{\overline{s}}$. We remark that the two types of thresholds $v_{\underline{s}}$ and $v_{\overline{s}}$ are useful for different ways of boosting as specified in the remainder of this section.

Before stating the theorem, we need to first describe what it means to "partially" boost. The results given in this section are based on the framework of *List PAC learning* (Brukhim et al., 2022; Charikar and Pabbaraju, 2023). The connection between list learning and multiclass boosting was demonstrated by prior works (Brukhim et al., 2023a,b). Here we strengthen this link and generalize previous results to hold for arbitrary costs. We start with introducing list learning in Definition 40, followed by the statement of Theorem 41.

Definition 40 (**s-List PAC Learning (Brukhim et al., 2022; Charikar and Pabbaraju, 2023)**) *An algorithm $\mathcal{L}$ is a s-list PAC learner for a class $\mathcal{F} \subseteq \mathcal{Y}^{\mathcal{X}}$ if the following holds. For every distribution $\mathcal{D}$ over $\mathcal{X}$ and target function $f \in \mathcal{F}$, and every $\epsilon, \delta > 0$, if S is a set of $m \geq m(\epsilon, \delta)$ examples drawn i.i.d. from $\mathcal{D}$ and labeled by f, then $\mathcal{L}(S)$ returns $\mu : \mathcal{X} \to \mathcal{Y}^s$ such that, with probability at least $1 - \delta$,*

$$L_{\mathcal{D}}(\mu) \triangleq \mathbb{P}_{x\sim\mathcal{D}}\big[f(x) \notin \mu(x)\big] \;\leq\; \epsilon\,.$$

Theorem 41 *Let $\mathcal{Y} = [k]$, let $w : \mathcal{Y}^2 \to [0,1]$ be any cost function, and let $z \geq 0$. Let $v_{\underline{2}} \leq \ldots \leq v_{\underline{k}}$ as defined in Equation (67), and let $s < k$ be the smallest integer for which $z < v_{\underline{s+1}}$ or, if none exists, let $s = k$. Then, the following claims both hold.*

- (w, z) ***is*** s***-boostable:*** *for every class $\mathcal{F} \subseteq \mathcal{Y}^{\mathcal{X}}$, every (w, z)-learner is boostable to an s-list PAC learner.*
- (w, z) ***is not*** $(s-1)$***-boostable:*** *for some $\mathcal{F} \subseteq \mathcal{Y}^{\mathcal{X}}$ there exists a (w, z)-learner that cannot be boosted to an s'-list PAC learner with $s' < s$.*

In words, Theorem 41 implies that there is a partition of the loss values interval $[0, 1]$ into k sub-intervals or "buckets", based on $v_{\underline{s}}$. Then, any loss value z in a certain bucket $v_{\underline{s}}$ can be boosted to a list of size s, but not below that. In Theorem 44 we show the converse: that any s-list learner can be boosted to (w, z)-learner where z may be the lowest value within the s-th interval $v_{\overline{s}}$, but not below it.

The first item of Theorem 41 holds trivially for $z \geq v_{\underline{k}}$, since in that case $s = k$ and a k-list learner always exists. The case $z < v_{\underline{k}}$ is addressed by Lemma 42 below.

Lemma 42 *Let $\mathcal{Y} = [k]$, let $w : \mathcal{Y}^2 \to [0,1]$ be a cost function, and define $v_{\underline{2}} \leq \ldots \leq v_{\underline{k}}$ as in Equation (67). Let $\mathcal{F} \subseteq \mathcal{Y}^{\mathcal{X}}$ and let $\mathcal{A}$ be a (w, z)-learner for $\mathcal{F}$ with sample complexity m_0 such that $z < v_{\underline{k}}$. Let s be the smallest integer in $\{1, \ldots, k-1\}$ such that $z < v_{\underline{s+1}}$ and let $\gamma = v_{\underline{s+1}} - z > 0$ be the margin. Fix any $f \in \mathcal{F}$, let $\mathcal{D}$ be any distribution over $\mathcal{X}$, and let $S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$ be a multiset of m examples given by i.i.d. points $x_1, \ldots, x_m \sim \mathcal{D}$ labeled by f. Finally, fix any $\delta \in (0,1)$. If Algorithm 3 is given S, oracle access to $\mathcal{A}$, and parameters $T = \left\lceil \frac{18 \ln(m)}{\gamma^2} \right\rceil$, $\eta = \sqrt{\frac{2 \ln(m)}{T}}$, $\widehat{m} = m_0\left(\frac{\gamma}{3}, \frac{\delta}{T}\right)$, and $\sigma = \frac{2}{3}\gamma$, then Algorithm 3 makes T calls to $\mathcal{A}$ and returns $\mu_S : \mathcal{X} \to \mathcal{Y}^s$ such that with probability at least $1 - \delta$:*

$$y_i \in \mu_S(x_i) \qquad \forall i = 1, \ldots, m \,.$$

Proof Fix any $f \in \mathcal{F}$ and any $\epsilon, \delta \in (0,1)$. Let $\mathcal{D}$ be any distribution over $\mathcal{X}$ and let $S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$ be a multiset of m examples given by i.i.d. points $x_1, \ldots, x_m \sim \mathcal{D}$ labeled by f. The first part of this proof follows similar steps as the one of Lemma 13. In particular, fix any $i \in [m]$ and observe that, again by the regret analysis of Hedge and by the definitions of η and T, Algorithm 3 guarantees

$$\frac{1}{T} \sum_{t=1}^{T} w(h_t(x_i), y_i) \leq \frac{\gamma}{3} + \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}_{j \sim \mathcal{D}_t} \Big[w(h_t(x_j), y_j) \Big] \,. \tag{68}$$

Furthermore, since $\mathcal{A}$ is a (w, z)-learner for $\mathcal{F}$, and given the choices of $\widehat{m}$ and $h_t = \mathcal{A}(S_t)$, by a union bound over $t \in [T]$ we obtain that

$$\frac{1}{T} \sum_{t=1}^{T} \mathbb{E}_{j \sim \mathcal{D}_t} \Big[w(h_t(x_j), y_j) \Big] \leq z + \frac{\gamma}{3} \tag{69}$$

with probability at least $1 - \delta$.

Now, we show that the list predictor μ_S built by Algorithm 3 is consistent with S with sufficiently large probability. Precisely, by conditioning on the event given by the inequality in Equation (69), we now demonstrate that $y_i \in \mu_S(x_i)$ for every $i \in [m]$. Consider the function F :

$\mathcal{X} \times \mathcal{Y} \to [0,1]$ as defined by Algorithm 3. Then, by Equations (68) and (69) together with the definition of $\gamma > 0$, we obtain for every $i \in [m]$ that

$$\sum_{\ell \in \mathcal{Y}} F(x_i, \ell) \cdot w(\ell, y_i) = \frac{1}{T} \sum_{t=1}^{T} w(h_t(x_i), y_i) \leq z + \frac{2}{3}\gamma < z + \gamma = v_{\underline{s+1}} , \tag{70}$$

which in turn implies that $y_i \in \mu_S(x_i)$ by construction of μ_S.

We now proceed with a deterministic characterization of the lists returned by μ_S, independently of any conditioning. Fix any $x \in \mathcal{X}$. Let $\boldsymbol{p}^x \in \Delta_{\mathcal{Y}}$ be such that $p^x_\ell = F(x, \ell)$ for any $\ell \in \mathcal{Y}$, and observe that the sum involved within the condition for $y \in \mathcal{Y}$ in the construction of the list $\mu_S(x)$ is equal to $w(\boldsymbol{p}^x, y)$. Furthermore, it must be true that $s < k$ since $\gamma > 0$. We can then show that the list $\mu_S(x)$ satisfies

$$v_{\underline{s+1}} > \max_{y \in \mu_S(x)} w(\boldsymbol{p}^x, y) = \max_{\boldsymbol{q} \in \Delta_{\mu_S(x)}} w(\boldsymbol{p}^x, \boldsymbol{q}) \geq \min_{\boldsymbol{p} \in \Delta_{\mathcal{Y}}} \max_{\boldsymbol{q} \in \Delta_{\mu_S(x)}} w(\boldsymbol{p}, \boldsymbol{q}) = \mathrm{V}_{\mu_S(x)}(w) , \tag{71}$$

where the first inequality holds by definition of $\mu_S(x)$. Consequently, after observing that $\mathrm{V}_J(w) \leq \mathrm{V}_{J'}(w)$ if $J \subseteq J'$, note that

$$\mathrm{V}_{\mu_S(x)}(w) < v_{\underline{s+1}} = \min_{J \in \binom{\mathcal{Y}}{s+1}} \mathrm{V}_J(w) = \min_{J \subseteq \mathcal{Y} : |J| \geq s+1} \mathrm{V}_J(w) , \tag{72}$$

which in turn implies that $|\mu_S(x)| \leq s$. We can thus assume that μ_S outputs elements of $\mathcal{Y}^s$ without loss of generality. ■

The following lemma proves that, in a similar way as Lemma 14 for the binary setting, the list predictor output by Algorithm 3 has a sufficiently small generalization error provided that the sample S has size m sufficiently large. The main idea to prove generalization is again via a sample compression scheme, but relying instead of novel results by Brukhim et al. (2023a) for multiclass classification. This generalization result, combined with Lemma 42, suffices to prove Theorem 41.

Lemma 43 *Assume the hypotheses of Lemma 42. For any $\epsilon, \delta \in (0,1)$, if the size m of the sample given to Algorithm 3 satisfies*

$$m \geq \frac{\ln(2/\delta)}{\epsilon} + m_0\left(\frac{\gamma}{3}, \frac{\delta}{2T}\right) \cdot T \cdot \left(1 + \frac{\ln(m)}{\epsilon}\right) ,$$

then with probability at least $1-\delta$ the output μ_S of Algorithm 3 satisfies $L_{\mathcal{D}}(\mu_S) \leq \epsilon$. Therefore, Algorithm 3 is an s-list PAC learner for $\mathcal{F}$. Moreover, one can ensure that Algorithm 3 makes $O\big(\frac{\ln(m)}{\gamma^2}\big)$ calls to $\mathcal{A}$ and has sample complexity $m(\epsilon, \delta) = \widetilde{O}\left(m_0\left(\frac{\gamma}{3}, \frac{\delta}{2T}\right) \cdot \frac{\ln(1/\delta)}{\epsilon \cdot \gamma^2}\right)$.

Proof By analogy with the proof of Lemma 14, we first apply Lemma 42 with $\delta/2$ in place of δ in order to obtain an s-list predictor μ_S consistent with S with probability at least $1 - \delta/2$. We then apply a compression-based generalization argument. To do so, we remark once again that one can construct a compression scheme for μ_S of size equal to the total size of the samples on which $\mathcal{A}$ operates, which is equal to $\kappa = \widehat{m} \cdot T$. The main difference with the binary case is that we rely on the generalization bound for a sample compression scheme for lists as per Theorem 6 of Brukhim et al.

(2023a), with $\delta/2$ in place of δ; we can employ this generalization bound thanks to the consistency of μ_S with S (with sufficiently large probability). Then, this implies that

$$L_{\mathcal{D}}(\mu_S) = \mathbb{P}_{x\sim\mathcal{D}}\big[f(x) \notin \mu_S(x)\big] \leq \frac{\kappa \ln(m) + \ln(2/\delta)}{m-\kappa} \tag{73}$$

holds with probability at least $1-\delta/2$. By similar calculations as in Equations (18) and (19), and replacing the values of κ and $\widehat{m}$, the right-hand side of the above inequality can be easily show to be at most ϵ whenever:

$$m \geq \frac{\ln(2/\delta)}{\epsilon} + m_0\left(\frac{\gamma}{3}, \frac{\delta}{2T}\right) \cdot T \cdot \left(1 + \frac{\ln(m)}{\epsilon}\right) \ . \tag{74}$$

A union bound concludes the proof for the first part of the claim, since it shows that Algorithm 3 is an s-list PAC learner for $\mathcal{F}$. The sample complexity of Algorithm 3 then immediately follows by definition of T. ■

Next, we prove a kind of converse of Theorem 41. That is, one can convert a list learner to a weak learner. This results is stated formally as follows.

Theorem 44 (s-List $\Rightarrow$ Weak Learning) *There exists an algorithm $\mathcal{B}$ that for every $\mathcal{F} \subseteq \mathcal{Y}^{\mathcal{X}}$ satisfies what follows. Let $w : \mathcal{Y}^2 \to [0,1]$ be a cost function, let $z \geq 0$, and let $v_{\overline{2}} \leq \ldots \leq v_{\overline{k}}$ as defined in Equation (67). Let $s < k$ be the smallest integer for which $z < v_{\overline{s+1}}$, or if none exists let $s = k$. Given oracle access to a s-list PAC learner $\mathcal{L}$ for $\mathcal{F}$ with sample complexity $m_{\mathcal{L}}$, algorithm $\mathcal{B}$ is a (w,z)-learner for $\mathcal{F}$ with sample complexity $m_{\mathcal{L}}$ that makes one oracle call to $\mathcal{L}$.*

Proof Fix any $f \in \mathcal{F}$ and any distribution $\mathcal{D}$ over $\mathcal{X}$, and suppose $\mathcal{B}$ is given a sample S of size $m \geq m_{\mathcal{L}}(\epsilon,\delta)$ consisting of examples drawn i.i.d. from $\mathcal{D}$ and labeled by f. First, $\mathcal{B}$ calls $\mathcal{L}$ on S. By hypothesis, then, $\mathcal{L}$ returns $\mu_S : \mathcal{X} \to \mathcal{Y}^s$ that with probability at least $1-\delta$ satisfies:

$$\mathbb{P}_{x\sim\mathcal{D}}\big[f(x) \notin \mu_S(x)\big] \leq \epsilon \ . \tag{75}$$

Conditioning on the event above, we give a randomized predictor h_S such that $L^w_{\mathcal{D}}(h_S) \leq z + \epsilon$. First, for any nonempty $J \subseteq \mathcal{Y}$ let $\boldsymbol{p}_J \in \Delta_{\mathcal{Y}}$ be the minimax distribution achieving the value of the game restricted to J,

$$\boldsymbol{p}_J = \arg\min_{\boldsymbol{p}\in\Delta_{\mathcal{Y}}} \left(\max_{\boldsymbol{q}\in\Delta_J} w(\boldsymbol{p},\boldsymbol{q})\right) \ . \tag{76}$$

Then, simply define $h_S(x) \sim \boldsymbol{p}_{\mu_S(x)}$.

Let us analyse the loss of h_S under $\mathcal{D}$. First, by the law of total probability, and since $\|w\|_\infty \leq 1$,

$$L^w_{\mathcal{D}}(h_S) \leq \mathbb{P}_{x\sim D}[f(x) \notin \mu_S(x)] + \sum_J \mathbb{P}_{x\sim\mathcal{D}}[\mu_S(x) = J \wedge f(x) \in J] \cdot L^w_{\mathcal{D}_J}(h_S) \ , \tag{77}$$

where the summation is over all $J \subseteq \mathcal{Y}$ with $\mathbb{P}_{x\sim\mathcal{D}}[\mu_S(x) = J] > 0$, and $\mathcal{D}_J$ is the distribution obtained from $\mathcal{D}$ by conditioning on the event $\mu_S(x) = J \wedge f(x) \in J$. Consider the right-hand side of Equation (77). By Equation (75), the first term is at most ϵ. For the second term, denote by $\boldsymbol{q}_J$ the marginal of $\mathcal{D}_J$ over $\mathcal{Y}$; note that, crucially, $\boldsymbol{q}_J \in \Delta_J$. Therefore, by definition of h_S and of s:

$$L^w_{\mathcal{D}_J}(h_S) = w(\boldsymbol{p}_J, \boldsymbol{q}_J) \leq \max_{\boldsymbol{q}\in\Delta_J} w(\boldsymbol{p}_J, \boldsymbol{q}) = \mathrm{V}_J(w) \leq v_{\overline{s}} \leq z \ . \tag{78}$$

We conclude that $L^w_{\mathcal{D}}(h_S) \leq z + \epsilon$. ■

C.4. Lower bounds

We start proving the second item of Theorem C.

Lemma 45 *Let $\mathcal{Y} = [k]$ and let $w : \mathcal{Y}^2 \to [0,1]$ be any cost function. Let $0 = v_1(w) \leq v_2(w) \leq \cdots \leq v_\tau(w)$ as defined in Equation (6), and let n be the largest integer for which $v_n(w) \leq z$. Then there exists a class $\mathcal{F} \subseteq \mathcal{Y}^\mathcal{X}$ that has a (w,z)-learner but no (w,z')-learner for any $z' < v_n(w)$.*

Proof If $v_n(w) = 0$ then the claim is trivial, hence we assume $v_n(w) > 0$. Observe that this implies that there exists some J with $|J| \geq 2$ such that $v_n(w) = V_J(w)$. Let $\boldsymbol{p}^* \in \Delta_\mathcal{Y}$ be the distribution achieving $\mathrm{V}_J(w)$, that is:

$$\boldsymbol{p}^* = \arg\min_{\boldsymbol{p}\in\Delta_\mathcal{Y}} \max_{\boldsymbol{q}\in\Delta_J} w(\boldsymbol{p},\boldsymbol{q}) \ . \tag{79}$$

Now let $\mathcal{X}$ be any infinite domain (e.g., $\mathcal{X} = \mathbb{N}$) and let $\mathcal{F} = J^\mathcal{X}$. For every distribution $\mathcal{D}$ over $\mathcal{X}$ and every labeling $f : \mathcal{X} \to J$,

$$L^w_\mathcal{D}(\boldsymbol{p}^*) = w(\boldsymbol{p}^*, \boldsymbol{q}_\mathcal{D}) \leq \max_{\boldsymbol{q}\in\Delta_J} w(\boldsymbol{p}^*,\boldsymbol{q}) = \mathrm{V}_J(w) = v_n(w) \leq z \ , \tag{80}$$

where $\boldsymbol{q}_\mathcal{D} \in \Delta_J$ is the marginal of $\mathcal{D}$ over $\mathcal{Y}$. Thus the trivial learner that returns the random guess h based on $\boldsymbol{p}^*$ is a (w,z)-learner for $\mathcal{F}$.

Next, we show that $\mathcal{F}$ has no (w,z')-learner with $z' < v_n(w)$. Suppose indeed towards a contradiction that there exists such a (w,z')-learner $\mathcal{A}$. By Theorem 32, then, $\mathcal{F}$ admits a (w,z')-list learner $\mathcal{L}$. Now, as $z' < v_n = \mathrm{V}_J(w)$, the list function μ returned by $\mathcal{L}$ ensures $J \not\subseteq \mu(x)$ for all $x \in \mathcal{X}$. Moreover, as $F = J^\mathcal{X}$, we can assume without loss of generality that $\mu(x) \subseteq J$ for all $x \in \mathcal{X}$; otherwise we could remove the elements of $\mu(x) \setminus J$ to obtain a list $\mu'(x)$ such that $\mathrm{V}_{\mu'(x)}(w) \leq \mathrm{V}_{\mu(x)}(w)$ and that $\mathbb{P}_{x\sim\mathcal{D}}[f(x) \in \mu'(x)] \geq \mathbb{P}_{x\sim\mathcal{D}}[f(x) \in \mu(x)]$. It follows that $\mu(x) \subsetneq J$ for all $x \in \mathcal{X}$. Therefore, $\mathcal{L}$ is a $(|J|-1)$-list PAC learner. However, one can verify that the $(|J|-1)$-Daniely-Shalev-Shwartz dimension of $\mathcal{F}$ is unbounded (see Definition 6 of Charikar and Pabbaraju (2023)). It follows by Charikar and Pabbaraju (2023, Theorem 1) that $\mathcal{F}$ is not $(|J|-1)$-list PAC learnable, yielding a contradiction. ■

Next, we prove the second item of Theorem 41.

Lemma 46 *Let $\mathcal{Y} = [k]$, let $w : \mathcal{Y}^2 \to [0,1]$ be any cost function, and let $v_{\underline{2}} \leq \ldots \leq v_{\underline{k}}$ as defined in Equation (67). Let $z \geq v_{\underline{2}}$ and let $s \leq k$ be the largest integer for which $z \geq v_{\underline{s}}$. Then, there exists a class $\mathcal{F} \subseteq \mathcal{Y}^\mathcal{X}$ that has a (w,z)-learner and is not $(s-1)$-list PAC learnable.*

Proof The proof follows closely the one of Lemma 45. Let $J = \arg\min\left\{\mathrm{V}_{J'}(w) : J' \in \binom{\mathcal{Y}}{s}\right\}$. Let $\mathcal{X}$ be any infinite domain (e.g., $\mathcal{X} = \mathbb{N}$) and let $\mathcal{F} = J^\mathcal{X}$. As in the proof of Lemma 45 there is a distribution $\boldsymbol{p}^*$ yielding a (w,z)-learner for $\mathcal{F}$. Simultaneously (again from the proof of Lemma 45), the class $\mathcal{F}$ is not $(s-1)$-list PAC learnable. ■

Finally, we prove the second item of Theorem D.

Lemma 47 *Let $\mathcal{Y} = [k]$, let $\boldsymbol{w} = (w_1, \ldots, w_r)$ where each $w_i : \mathcal{Y}^2 \to [0,1]$ is a cost function, and let $\boldsymbol{z}, \boldsymbol{z}' \in [0,1]^r$ such that $\boldsymbol{z} \not\preceq_{\boldsymbol{w}} \boldsymbol{z}'$. There exists a class $\mathcal{F} \subseteq \mathcal{Y}^\mathcal{X}$ that admits a $(\boldsymbol{w},\boldsymbol{z})$-learner that is trivial, and therefore cannot be boosted to a $(\boldsymbol{w},\boldsymbol{z}')$-learner.*

Proof By assumption that $\boldsymbol{z} \not\preceq_{\boldsymbol{w}} \boldsymbol{z}'$, there must be a set $J \subseteq \mathcal{Y}$ for which $\boldsymbol{z} \in D_J(\boldsymbol{w})$ and $\boldsymbol{z}' \notin D_J(\boldsymbol{w})$. Let $\mathcal{X}$ be any infinite domain (e.g., $\mathcal{X} = \mathbb{N}$) and let $\mathcal{F} = J^{\mathcal{X}}$. By definition of $D_J(\boldsymbol{w})$ there exists a trivial learner that is a $(\boldsymbol{w}, \boldsymbol{z})$-learner for $\mathcal{F}$. Assume towards contradiction that the trivial learner can be used to construct a $(\boldsymbol{w}, \boldsymbol{z}')$-learner for $\mathcal{F}$. Since $\boldsymbol{z}' \notin D_J(\boldsymbol{w})$, then by Proposition 27 there exists $\boldsymbol{\alpha} \in \Delta_r$ such that $\langle \boldsymbol{\alpha}, \boldsymbol{z}' \rangle < \mathrm{V}_J(w_{\boldsymbol{\alpha}})$, where $w_{\boldsymbol{\alpha}} = \sum_{i=1}^r \alpha_i w_i$. Then, by Theorem 32, there is a $(w_{\boldsymbol{\alpha}}, z'_{\boldsymbol{\alpha}})$-list learner $\mathcal{L}'$ that outputs list functions μ such that, for every $x \in \mathcal{X}$ it holds that:

$$V_{\mu(x)}(w_{\boldsymbol{\alpha}}) \leq z'_{\boldsymbol{\alpha}} < \mathrm{V}_J(w_{\boldsymbol{\alpha}}).$$

Thus $\mathcal{L}'$ is a $(|J|-1)$-list PAC learner for $\mathcal{F}$. However, one can verify that the $(|J|-1)$-Daniely-Shalev-Shwartz dimension of $\mathcal{F}$ is unbounded (see Definition 6 of Charikar and Pabbaraju (2023)). It follows by Charikar and Pabbaraju (2023, Theorem 1) that $\mathcal{F}$ is not $(|J|-1)$-list PAC learnable, yielding a contradiction. ■

Appendix D. Cost-Sensitive Learning

Cost sensitive learning has a long history in machine learning. The two main motivations are that, first, not all errors have the same impact, and second, that there might be a class-imbalance between the frequency of different classes. Over the years there have been many workshops on cost-sensitive learning in ICML (2000), NIPS (2008), SDM (2018), yet another indication of the importance of cost-sensitive loss. See, e.g., Ling and Sheng (2017, 2008). Recall the definition of the cost sensitive loss:

$$L_{\mathcal{D}}^{w}(h) = \mathbb{E}_{x\sim\mathcal{D}}\Big[w(h(x), f(x))\Big]\,. \tag{81}$$

Given the the distribution $\mathcal{D}$ on $\mathcal{X}$, the Bayes optimal prediction rule is given by:

$$\arg\min_{i\in\mathcal{Y}} \sum_{j\in\mathcal{Y}} w(i,j)\,\mathbb{P}[f(x)=j \mid x] \tag{82}$$

for every $x \in \mathcal{X}$. The early works include Elkan (2001), which characterizes the Bayes optimal predictor as a threshold, and its implication for re-balancing and decision tree learning. In fact, this resembles our binary prediction rule.

Sample complexity for cost-sensitive leaning for large margin appears in Karakoulas and Shawe-Taylor (1998). Additional sample complexity bounds, for cost-sensitive learning, based on transformation of the learning algorithms and using rejection sampling is found in Zadrozny et al. (2003). The idea of cost sensitive learning has a long history in statistics. For binary classification, there are many different metrics used for evaluation. Let $w = \left(\begin{smallmatrix} w_{++} & w_{+-} \\ w_{-+} & w_{--}\end{smallmatrix}\right)$. The false-positive (FP) is w_{+-}, the false-negative is w_{-+}, the precision is $w_{++}/(w_{++}+w_{+-})$, the recall is $w_{++}/(w_{++}+w_{-+})$, and more.

D.1. Boosting cost-sensitive loss

There has been significant amount of work with the motivation of adapting AdaBoost to a cost-sensitive loss. At a high level, the different proposals either modify the way the algorithm updates its weights, taking in to account the cost-sensitive loss, or changes the final prediction rule. Nikolaou et al. (2016) give an overview on various AdaBoost variants for the cost-sensitive case. See also

Landesa-Vázquez and Alba-Castro (2015). Our modified update rule of Hedge in Algorithm 1 corresponds to CGAda of Sun et al. (2007) and related AdaBoost variants.

The theoretically sound proposal focused on deriving similar guarantees to that of AdaBoost. Cost-sensitive boosting by Masnadi-Shirazi and Vasconcelos (2011) modified the exponential updates to include the cost-sensitive loss. Cost-Generalized AdaBoost by Landesa-Vazquez and Alba-Castro (2012) modifies the initial distribution over the examples. AdaboostDB by Landesa-Vazquez and Alba-Castro (2013) modifies the base of the exponents used in the updates. All the theoretically sound proposal are aimed to guarantee convergence under the standard weak-leaning assumption. Their main objective is to derive better empirical results when faced with a cost-sensitive loss. However, they do not address the essential question, *when is boosting possible?* In particular, they do not study cost-sensitive variants weak learners and do not characterize the boostability of such learners. In addition to the papers above, there have been many heuristic modifications of AdaBoost which try to address the cost-sensitive loss (Karakoulas and Shawe-Taylor, 1998; Fan et al., 1999; Ting, 2000; Viola and Jones, 2001; Sun et al., 2007).

D.2. Multi-objective learning and boosting

Learning with multiple objectives is a common topic in machine learning (Jin, 2007), and scalarization techniques are a popular tool used to reduce the problem to standard single-objective learning (Jin and Sendhoff, 2008; Lin et al., 2019; Hu et al., 2024). Our results show that multi-objective cost-sensitive learning is indeed equivalent to learning with respect to any possible scalarization of the multi-objective loss. Multi-objective learning is also studied in online settings (Jiang et al., 2023) and multi-armed bandit settings Lu et al. (2019). Game-theoretic tools and no-regret analysis, which our analysis relies upon, are also ubiquitous in multicalibration (Haghtalab et al., 2024).

Well-studied special cases of multi-objective learning are variants of learning with *one-sided error* (Kivinen, 1995; Sabato and Tishby, 2012). A typical goal in one-sided learning or the related *positive-reliable* learning (Kalai et al., 2012; Kanade and Thaler, 2014; Durgin and Juba, 2019) is to guarantee an (almost) 0 false positive loss and simultaneously a low false negative loss. This corresponds to a $(\boldsymbol{w}, \boldsymbol{z})$-learner with $\boldsymbol{w} = (w_+, w_-)$ and $\boldsymbol{z} = (0, \epsilon)$. More generally, Kalai et al. (2012) also considered *tolerant reliable* learning with an arbitrary $\boldsymbol{z} = (z_+, z_-)$. Our results apply in this context. For example in the binary case, our results show that a $(\boldsymbol{w}, \boldsymbol{z})$-learner (i.e., a $\boldsymbol{z}$-tolerant-reliable one) is boostable if and only if $\sqrt{z_+} + \sqrt{z_-} < 1$. Moreover, our results also imply boostability and learnability results for reliable learning in the multi-class case; a learning setting mostly over-looked so far.

Appendix E. Multiclass Boosting

Boosting is a fundamental methodology in machine learning that can boost the accuracy of weak learning rules into a strong one. Boosting theory was originally designed for binary classification. The study of boosting was initiating in a line of seminal works which include the celebrated Adaboost algorithm, as well an many other algorithms with various applications, (see e.g. Kearns (1988); Schapire (1990); Freund (1990); Freund and Schapire (1997)). It was later adapted to other settings and was extensively studied in broader contexts as well (Ben-David et al., 2001; Kalai and Servedio, 2005; Long and Servedio, 2008; Kalai et al., 2008; Kanade and Kalai, 2009; Feldman, 2009; Møller Høgsgaard et al., 2024; Green Larsen and Ritzert, 2022; Raman and Tewari, 2022; Raman et al., 2019).

There are also various extension of boosting to the multiclass setting. The early extensions include AdaBoost.MH, AdaBoost.MR, and approaches based on Error-Correcting Output Codes (ECOC) (Schapire and Singer, 1999; Allwein et al., 2000). These works often reduce the k-class task into a single binary task. The binary reduction can have various problems, including increased complexity, and lack of guarantees of an optimal joint predictor.

Notably, a work by Mukherjee and Schapire (2011) established a theoretical framework for multiclass boosting, which generalizes previous learning conditions. However, this requires the assumption that the weak learner minimizes a complicated loss function, that is significantly different from simple classification error.

More recently, there have been several works on multiclass boosting. First, Brukhim et al. (2021) followed a formulation for multiclass boosting similar to that of Mukherjee and Schapire (2011). They proved a certain hardness result showing that a broad, yet restricted, class of boosting algorithms must incur a cost which scales polynomially with $|\mathcal{Y}|$. Then, Brukhim et al. (2023b) and Brukhim et al. (2023a) demonstrated efficient methods for multiclass boosting. We note that our work generalizes the results given in Brukhim et al. (2023a) to the cost sensitive setting, as detailed in Appendix C.

Appendix F. Discussion on Randomized Predictors

Recall that we see randomized predictors as functions $h : \mathcal{X} \to \Delta_{\mathcal{Y}}$ such that $h(x)$ is some distribution $\boldsymbol{p}^x \in \Delta_{\mathcal{Y}}$ for every $x \in \mathcal{X}$. We may instead assume that randomized predictors are randomized procedures h' that behave as follows: for every $x \in \mathcal{X}$, every invocation of $h'(x)$ returns a fresh, independent random label distributed according to $\boldsymbol{p}^x$. Note that, given access to h', one can estimate $\boldsymbol{p}^x = h(x)$ for any given $x \in \mathcal{X}$ within total variation distance ϵ with probability at least $1 - \delta$ by performing $\mathcal{O}\left(\frac{|\mathcal{Y}|+\ln(1/\delta)}{\epsilon^2}\right)$ calls to $h'(x)$; see Canonne (2020, Theorem 1). All our algorithms and bounds can therefore be adapted so as to employ only the predictors h'. We remark that this does not affect the sample complexity of our algorithms (but it may affect their running time).