Faster Recalibration of an Online Predictor via Approachability

Princewill Okoroafor
Cornell University

Wen Sun
Cornell University

Robert Kleinberg
Cornell University

Abstract

Predictive models in ML need to be trustworthy and reliable, which often at the very least means outputting calibrated probabilities. This can be particularly difficult to guarantee in the online prediction setting when the outcome sequence can be generated adversarially. In this paper we introduce a technique using Blackwell's approachability theorem for taking an online predictive model which might not be calibrated and transforming its predictions to calibrated predictions without much increase to the loss of the original model. Our proposed algorithm achieves calibration and accuracy at a faster rate than existing techniques (Kuleshov and Ermon, 2017) and is the first algorithm to offer a flexible tradeoff between calibration error and accuracy in the online setting. We demonstrate this by characterizing the space of jointly achievable calibration and regret using our technique.

1 Introduction

In the online learning setting, a predictive model, also known as a forecaster, gives a probability value prediction at each time step, and its performance is evaluated based on a loss function. For the class of loss function known as a proper scoring rule, the only way to minimize that score is to predict the true probabilities of an outcome. For most prediction problems we do not know how to compute the true probabilities of outcomes, and the best we can do is to use a trained model (e.g., a deep neural network or contextual bandit algorithm) attaining a low scoring-rule loss without necessarily minimizing it. However, most training methods for predictive models do not guarantee calibrated probability values. There has been a large body of work highlighting the need for calibrated probability estimates (i.e., models that are able to assess their uncertainty) (Jiang et al., 2011; DeGroot and Fienberg, 1983) and on how to obtain these calibrated probability estimates Foster (1999). In the offline setting, this is generally done by some post-processing of the data to remap the probability values to calibrated probability estimates in a way that minimizes the increase in loss, such as by post-hoc calibration or recalibration. In contrast, in the online prediction setting, little work has been done on this subject. Recently, Kuleshov and Ermon (2017) and Foster and Hart (2021) have presented various approaches for taking an online predictive model and transforming its predictions without major increase in loss. Kuleshov and Ermon (2017) introduced this problem as an online recalibration problem, and provided an algorithm for achieving epsilon accuracy relative to the loss function using a connection between calibration and internal regret. In this paper, we show that their result can be significantly improved by using Blackwell's Approachability Theorem. We present an algorithm, making use of approachability, that achieves recalibration at a much faster rate than the internal regret minimization algorithm by Kuleshov and Ermon (2017). We also characterize the achievable amount of calibration and regret as a function of the time horizon using our technique; more precisely, we study for which exponents a, b does there exist a forecasting algorithm that guarantees at most T^a calibration error and no more than T^b regret relative to scoring rule loss functions. We provide the first algorithm that offers a flexible tradeoff between calibration error and regret in the online setting.

1.1 Motivation

Calibrating probability predictions As the prevalence of machine learning systems in decision-making settings grows, it is essential that the predictions they provide are trustworthy, especially in applications where the confidence associated with the prediction is at least as important as the prediction itself. Neural networks have been found to be poor at assessing their own uncertainty (Guo et al., 2017), and as a result,

Proceedings of the 27[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2024, Valencia, Spain. PMLR: Volume 238.

may output probability values that do not match the true probabilities of outcomes. This can have serious consequences; machine learning systems have been known to propagate unintended but harmful discrimination, as shown by Buolamwini and Gebru (2018) for image classification and Bolukbasi et al. (2016) for natural language tasks. One proposed method for addressing the issue of assessing uncertainty is calibration (Hebert-Johnson et al., 2018). Calibration requires that the probability estimates from the ML model match the true distribution of the outcome; for example, for a binary class, if a model outputs a probability of 0.3 a certain number of times, the proportion of true outcomes should be 30 percent across the total instances when the model predicted 0.3. In the online setting, many works have proposed techniques for how to achieve calibrated probability estimates, even in the adversarial setting (Foster, 1999; Mannor and Stoltz, 2010; Abernethy et al., 2011).

Limitations of calibration While calibration is a useful property for online predictors to have, calibration is not sufficient and does not fully reflect domain specific knowledge. For example, consider two ML weather forecasters. Suppose the true outcome is that it rains once every two days. Forecaster 1 predicts 50 percent chance of rain every day, and Forecaster 2 predicts 0 percent chance of rain on the days it does not rain and 100 percent on the days it does. Observe that both of these forecasters are equally calibrated; however, the second forecaster is a better predictor of the likelihood of rain. Calibration does not capture this fact. Although calibration does not imply accuracy, accuracy does imply calibration, simply because being accurate requires an understanding of the outcome distribution. This is why, in practice, proper scoring rules are used to assess the accuracy of predictions (Gneiting and Raftery, 2007).

Incorporating expert/domain-specific knowledge in online prediction models Forecaster 2 is an example of a forecaster that reflects domain-specific knowledge and is also calibrated. However, it is also possible for a forecaster that acts on domain specific knowledge to be poorly calibrated. Consider a third forecaster in the same weather prediction setting which predicts 20 percent chance of rain on the days it does not rain, and 80 percent chance of rain on the days that it does. This predictor is poorly calibrated, because it incurs a calibration error of 0.2 for every decision. However, compared to Forecaster 1, its predictions still reflect a domain-specific understanding of the probability distribution. The goal of our work is to take a model such as this third forecaster and transform its predictions in an online setting to achieve calibration while still making decisions that are informed by domain knowledge.

1.2 Problem formulation

In this paper, we focus on a class of loss functions known as strictly proper scoring rules. We refer the reader to Section 2.2 for an introduction on the subject.

Consider an online prediction environment where the timing of each round of the prediction process is as follows.

1. An oracle reveals a prediction q_t.
2. The algorithm must make a prediction p_t.
3. The actual label $y_t \in \{0, 1\}$ is revealed.
4. The algorithm receives a score $S(p_t, y_t)$.

At the end of T rounds, the following quantities are calculated.

- The forecaster's cumulative score is $S_f = \sum_{t=1}^{T} S(p_t, y_t)$.
- The oracle's cumulative score is $S_o = \sum_{t=1}^{T} S(q_t, y_t)$.
- The forecaster's average regret is $\frac{1}{T}(S_f - S_o)$.
- The forecaster's ℓ_1-calibration error is

$$\sum_{p \in \{p_1, \ldots, p_T\}} \left| \tfrac{1}{T} \sum_{t=1}^{T} (y_t - p) \cdot \mathbf{1}_{p_t = p} \right|.$$

  (Although written as a sum over all $p \in [0, 1]$, the sum is actually finite because there are only finitely many p for which the summand is nonzero.)

For the sake of generality, our model makes no assumptions about how the oracle's predictions are generated, except that if the algorithm is randomized the oracle cannot anticipate the algorithm's *future* coin-tosses. This means, for example, that our simple prediction model subsumes more elaborate models in which the predictions q_t are generated by a contextual bandit algorithm, or by a pre-trained model such as a deep neural network, using domain-specific features observed at time t or earlier.

Our work addresses the question: for which exponent pairs (a, b) is there a forecasting algorithm that guarantees regret $\tilde{O}(T^a)$ and calibration error $\tilde{O}(T^b)$? The purpose of this paper is to propose a method of tackling this question using Blackwell's Approachability Theorem.

1.3 Our results

For the notion of regret described above, we show that there is a family of approachability-based algorithms, parameterized by $\varepsilon > 0$, that simultaneously achieves calibration $O(\varepsilon + 1/\sqrt{\varepsilon T})$ and average regret $O(\varepsilon^2 + 1/\sqrt{\varepsilon T})$. This is a significant improvement from the result by Kuleshov and Ermon (2017), which achieves calibration $O(\varepsilon + 1/\sqrt{\varepsilon^2 T})$ and average regret $O(\varepsilon + 1/\sqrt{\varepsilon^2 T})$. The improved dependence on ε is significant in practice because it impacts how many samples, T, are required in order to make the average regret less than some specified upper bound, δ. For example, to make $\varepsilon^2 + 1/\sqrt{\varepsilon T}$ less than δ one would set $T = O(\delta^{-5/2})$ and $\varepsilon = O(\delta^{1/2})$, whereas to make $\varepsilon + 1/\sqrt{\varepsilon^2 T}$ less than δ requires $T = O(\delta^{-4})$ and $\varepsilon = O(\delta)$. For $\delta = 0.1$ this amounts to the difference between a few hundred samples versus more than ten thousand.

By choosing ε appropriately, we show that our algorithm can be designed to achieve the best known calibration upper bound of $T^{-\frac{1}{3}}$ while limiting regret to no more than $T^{-\frac{1}{3}}$. If one is more interested in minimizing regret, we also show that ε can be chosen to achieve regret of $T^{-\frac{2}{5}}$ while limiting calibration error to no more than $T^{-\frac{1}{5}}$. The algorithm allows for a linear interpolation between these two bounds. That is, for any x in the interval $[\frac{1}{3}, \frac{2}{5}]$, we can set $\varepsilon = T^{-2x}$ to achieve calibration $O(T^{2x-1})$ while simultaneously achieving regret $O(T^{-x})$.

1.4 Comparison to prior work

Calibration and proper scoring rules Foster (1999) first reduced calibration to approachability. Since then, a number of alternative proofs of calibration using reductions to approachability have emerged (Mannor and Stoltz, 2010; Abernethy et al., 2011). Our work draws ideas and techniques from these papers, and extends those ideas with innovations specific to the task of online recalibration. Unlike in the standard setting of calibrated binary sequence prediction, the recalibration problem incorporates side information in the form of an oracle who makes a prediction at each timestep. In the standard calibration problem, the goal is to minimize calibration error. In the recalibration problem, the algorithm must attain two goals simultaneously: sublinear calibration error and sublinear regret relative to the oracle's predictions. To achieve both of these objectives we need to modify the vector payoffs and the approachable set used in the standard reduction from calibration to approachability. The main technical innovation in this work lies in verifying that the modified set is indeed approachable in the modified vector-payoff game. After showing that the modified set is indeed approachable, we rely on a reduction from approachability to Online Linear Optimization by Abernethy et al. (2011) to construct an algorithm for recalibration. The geometry of our approachable set leads to quantitative bounds on calibration error and regret that improve upon the state of the art.

Recalibration in offline setting In the offline setting, calibrated predictions are usually constructed using methods such as Platt Scaling (Platt, 1999) and isotonic regression (Niculescu-Mizil and Caruana, 2005). In the context of binary classification, these methods reduce the problem of outputting calibrated predictions to a one-dimensional regression problem. Given data $\{(x_i, y_i)\}_{i=1}^n$, they train a model $f(s)$ to predict $p_i = f(s)$ from uncalibrated scores $s_i = g(x_i)$ produced by a classifier g. These techniques are particularly suited for the offline setting where the training and the calibration phases of the algorithm can be separated and thus, do not apply in the online setting and can fail when the test distribution does not match the training distribution. Our results, on the other hand, are robust to adversarial manipulations.

Recalibration in online setting Kuleshov and Ermon (2017) present an algorithm for recalibration, that is, for achieving ε calibration and ε regret simultaneously at a rate of $1/\varepsilon\sqrt{T}$. They achieve this by running $1/\varepsilon$ many calibration algorithms in parallel for each prediction interval that the expert (called "oracle" in our work, "blackbox predictor" in theirs) makes. This method works because calibrated predictors have been shown to minimize internal regret (Cesa-Bianchi and Lugosi, 2006). They are able to bound the regret by the internal regret, which is bounded by calibration error, which itself is bounded by ε. The two main issues with their approach are first, the additional cost of running $1/\varepsilon$ calibration algorithms in parallel; and second, having to rely on the calibration error bound in order to bound the regret. Our technique bypasses these constraints by appealing to Blackwell's Approachability Theorem. With Blackwell's Approachability Theorem, we can treat this problem as a vector-valued game where one tries to simultaneously minimize the calibration and regret components of the vector. Instead of having $1/\varepsilon$ different calibration algorithms, we have only a single calibration algorithm which also takes regret into account. The single calibration algorithm achieves a stronger guarantee by leveraging the fact that proper scoring rules incentivize calibration. We also take this a step further by giving precise error bounds as a function of the time horizon, and allowing a trade-off between calibration error and regret.

Online Minimax multiobjective optimization An even more general problem than recalibration is online multiobjective optimization, for which Lee et al. (2022) present a minimax theorem and a multiplicative-weights algorithm that achieves (a suitable notion of) the minimax value plus a sublinear regret term. By casting recalibration as an online multiobjective optimization problem, we show in Appendix B how to achieve calibration error and average regret both bounded by $O(\varepsilon + 1/\sqrt{\varepsilon T})$, for any $\varepsilon > 0$. This matches the calibration error bound for our Algorithm 1, but with a worse dependence on ε in the regret bound. The reduction from recalibration to online multiobjective optimization uses loss vectors of dimension roughly $2^{1/\varepsilon}$, so a naïve implementation of the algorithm of Lee et al. (2022) would be computationally inefficient. In the Appendix, we indicate how it can be implemented to run in time $\text{poly}(1/\varepsilon)$ per iteration by exploiting the special structure of the loss vectors arising from our reduction. This running time is exponentially faster than the naïve reduction, but still exponentially slower than the $O(\log(1/\varepsilon))$ running time per iteration of our Algorithm 1.

Calibeating Another closely related result is contained in a preprint by Foster and Hart (2021). In their paper on "calibeating," they present a method for transforming expert predictions to calibrated predictions, while measuring accuracy against an even more strict benchmark than ours: they compare the algorithm's loss to that of the expert after the calibration error has been removed, a benchmark called the "refinement score". They prove this for the loss function known as the Brier score, when calibration is quantified using the ℓ_2 objective. Our result is incomparable to theirs: while their benchmark for accuracy is stricter than ours, our quantification of calibration (using ℓ_1 rather than ℓ_2) is stricter than theirs. Furthermore, our recalibration procedure applies to any strictly proper scoring rule loss, whereas their calibeating procedure is specialized to the Brier score.

2 Background

2.1 Calibration

Let $y_1, y_2, \ldots \in \{0, 1\}$ be a sequence of outcomes, and $p_1, p_2, \ldots \in [0, 1]$ a sequence of probability predictions by a forecaster. We define for every T and every pair p, ε where $0 \le p \le 1$ and $\varepsilon > 0$, the quantities

$$n_T(p, \varepsilon) := \sum_{t=1}^{T} \mathbb{I}[p_t \in (p - \varepsilon/2, p + \varepsilon/2)],$$

$$\rho_T(p, \varepsilon) := \frac{\sum_{t=1}^{T} y_t \mathbb{I}[p_t \in (p - \varepsilon/2, p + \varepsilon/2)]}{n_T(p, \varepsilon)}.$$

The quantity $\rho_T(p-\varepsilon/2, p+\varepsilon/2)$ should be interpreted as the empirical frequency of $y_t = 1$, up to round T, on only those rounds where the forecaster's prediction was "roughly" equal to p. The goal of calibration, of course, is to have this empirical frequency $\rho_T(p, \varepsilon)$ be close to the estimated frequency p. To capture how close an algorithm $\mathcal{A}$ to being ε-calibrated, we use a notion of rate below.

Definition 1. *Let $\mathcal{P}(\varepsilon)$ denote the set of midpoints of the intervals $[i\varepsilon, (i+1)\varepsilon]$ for $i = 0, 1, \ldots, \lfloor \varepsilon^{-1} \rfloor$. Let the (ℓ_1, ε)-calibration rate for forecaster $\mathcal{A}$ be*

$$C_T^{\varepsilon}(\mathcal{A}) = \max\left\{0,\ \frac{1}{T}\left(\sum_{z \in \mathcal{P}(\varepsilon)} n_T(z, \varepsilon) \cdot |z - \rho_T(z, \varepsilon)|\right) - \frac{\varepsilon}{2}\right\}$$

*We say that a forecaster is (ℓ_1, ε)-*calibrated *if $C_T^{\varepsilon}(\mathcal{A}) = o(1)$. This in turn implies $\limsup_{T\to\infty} C_T^{\varepsilon}(\mathcal{A}) = 0$.*

2.2 Proper Scoring Rules, Regret, and Recalibration

Kuleshov and Ermon (2017) define the problem of online recalibration in which the task is to transform a sequence of uncalibrated forecasts q_t into predictions p_t that are calibrated and almost as accurate as the original q_t. They show that this objective is achievable if and only if the loss function used to measure forecast accuracy is a *proper scoring rule*, a term which we now define.

Suppose there is a future event denoted by a random variable X with a finite set $\mathcal{Y}$ of possible outcomes. For example: $\mathcal{Y} = \{\text{rain}, \text{no rain}\}$. Let $\Delta_{\mathcal{Y}}$ be the set of probability distributions on $\mathcal{Y}$. An algorithm reports a probability distribution $p \in \Delta_{\mathcal{Y}}$, observes the outcome $y \in \mathcal{Y}$ and receives a score $S(p, y)$.

Definition 2. *A scoring rule is a function $S : \Delta_{\mathcal{Y}} \times \mathcal{Y} \mapsto \mathbb{R}$. It is proper if accurately reporting the distribution of X minimizes the expected score: that is, for all distributions $p, q \in \Delta_{\mathcal{Y}}$*

$$\mathbb{E}_{X\sim p}\left[S(p, X)\right] \le \mathbb{E}_{X\sim p}\left[S(q, X)\right]. \tag{1}$$

Scoring rule S is strictly proper *if Inequality (1) is strict whenever $p \neq q$.*

Note that we adopt the convention that the scoring rule is a loss function rather than a payoff function, i.e. p is the unique probability that minimizes $S(\cdot, p)$ rather than maximizing it. We extend S to the domain $\Delta_{\mathcal{Y}} \times \Delta_{\mathcal{Y}}$ by making it linear in the second variable. In other words, $S(q, p)$ is shorthand for $\mathbb{E}_{X\sim p}\left[S(q, X)\right]$.

We assume the scoring rule S is Lipschitz-continuous in its first variable, with Lipschitz constant L_S, i.e.

$$\forall p, q \in \Delta_{\mathcal{Y}} \; \forall y \in \mathcal{Y} \qquad |S(p,y) - S(q,y)| \leq L_S \cdot \|p - q\|,$$

where $\|p - q\|$ denotes the total variation distance between p and q.

We measure a forecaster's accuracy by comparing with the score of the oracle. Let $q_1, q_2, \ldots \in [0,1]$ be a sequence of probability predictions by an oracle.

Definition 3. *Let the regret at timestep t for forecaster $\mathcal{A}$ be*

$$r(p_t, q_t, y_t) = S(p_t, y_t) - S(q_t, y_t)$$

This leads to an average regret *of* $R_T(\mathcal{A}) = \frac{1}{T}\sum_{t=1}^{T} r(p_t, q_t, y_t)$. *We say that a forecaster has* no-regret *if* $R_T(\mathcal{A}) = o(1)$. *This in turn implies* $\limsup_{T\to\infty} R_T(\mathcal{A}) = 0$. *We also say a forecaster has* δ-regret rate *if* $R_T(\mathcal{A}) \leq \delta$.

Definition 4. *Let the $(\ell_1, \varepsilon, \delta)$-recalibration rate for forecaster $\mathcal{A}$ be*

$$C_T^{\varepsilon,\delta}(\mathcal{A}) = \max\left\{0, C_T^{\varepsilon}(\mathcal{A}), R_T(\mathcal{A}) - \frac{\delta}{2}\right\} \tag{2}$$

We say that a forecaster is $(\ell_1, \varepsilon, \delta)$-recalibrated *if* $C_T^{\varepsilon,\delta}(\mathcal{A}) = o(1)$. *This in turn implies* $\limsup_{T\to\infty} C_T^{\varepsilon,\delta}(\mathcal{A}) = 0$.

This definition is analogous to Definition 4 in Kuleshov and Ermon (2017), except that we have quantified the calibration and accuracy using two parameters, ε and δ, whereas they use ε for both.

2.3 Blackwell's Approachability Theorem

Blackwell approachability (Blackwell, 1956) generalizes the problem of playing a repeated two-player zero-sum game to games whose payoffs are vectors instead of scalars. In a Blackwell approachability game, at all times t, two players interact in this order: first, Player 1 selects an action $x_t \in X$; then, Player 2 selects an action $y_t \in Y$; finally, Player 1 incurs the vector-valued payoff $u(x_t, y_t) \in \mathbb{R}^d$. The sets X, Y of player actions are assumed to be compact convex subsets of finite-dimensional vector spaces, and u is assumed to be a biaffine function on $X \times Y$. Player 1's objective is to guarantee that the average payoff converges to some desired closed convex target set $\mathcal{S} \subseteq \mathbb{R}^d$. Formally, given target set $\mathcal{S} \subseteq \mathbb{R}^d$, Player 1's goal is to pick actions $x_1, x_2, \ldots \in X$ such that no matter the actions $y_1, y_2, \ldots \in Y$ played by Player 2,

$$\texttt{dist}\left(\frac{1}{T}\sum_{t=1}^{T} u(x_t, y_t), \mathcal{S}\right) \to 0 \quad \text{as} \quad T \to \infty \tag{3}$$

The action x_t is allowed to depend on the realized payoff vectors $u_s(x_s, y_s)$ for $s = 1, 2, \ldots, t-1$. We say the set S is approachable if Player 1 has a strategy that attains the goal (3) no matter how Player 2 plays. Blackwell's Approachability Theorem asserts that a convex set $\mathcal{S} \subset \mathbb{R}^d$ is approachable if and only if every closed halfspace containing $\mathcal{S}$ is approachable. Henceforth we refer to this necessary and sufficient condition as *halfspace-approachability*.

In this paper, we shall adopt the notation, $\texttt{dist}_p(x, \mathcal{S})$ to be the $\min_{s\in S} \|x - s\|_p$. We will refer to the ℓ_p ball $\in R^d$ of radius r centered at the origin as $B_p^d(r)$.

We now give an equivalent and alternative characterization of the definition of recalibration rate (Definition 4): let the *recalibration vector* at time T denoted $\mathbf{v}_T$ be given by: $\mathbf{v}_T = \mathbf{c}_T \oplus R_T$ where $\mathbf{c}_T(i) = \frac{n_T(i\varepsilon,\varepsilon)}{T}\left(i\varepsilon - \rho_T(i\varepsilon, \varepsilon)\right)$ for $0 \leq i \leq \lceil \varepsilon^{-1} \rceil$, and $R_T = \frac{1}{T}\sum_{t=0}^{T} S(p_t, y_t) - S(q_t, y_t)$.

Lemma 5.

$$C_T^{\varepsilon,\delta}(\mathcal{A}) = \max\left\{\texttt{dist}_1\left(\mathbf{c}_T, B_1^{\varepsilon^{-1}}(\varepsilon/2)\right), R_t - \delta/2\right\} \tag{4}$$

3 Recalibration via Approachability

We now describe the construction of the payoff game that allows us to reduce recalibration to approachability. This payoff game modifies the standard construction for calibration in (Foster, 1999; Abernethy et al., 2011) by adding an additional dimension for regret.

3.1 Reduction

For any $m \geq \sqrt{4L_S}$ where L_S is the lipschitz constant of the scoring rule, we will show how to construct an $(\ell_1, \varepsilon, \delta)$-recalibrated forecaster for $\varepsilon = \frac{1}{m}$ and $\delta = \frac{4L_S}{m^2}$. On each round t, after observing the oracle's prediction q_t, a forecaster will randomly predict a probability $p_t \in \{0/m, 1/m, 2/m, \ldots, (m-1)/m, 1\}$, according to the distribution $\mathbf{w}_t$, that is $\Pr(p_t = i/m) = w_t(i)$. We define a vector-valued game. Let the player choose $\mathbf{w}_t \in \mathcal{X} := \Delta_{m+1}$, and the adversary choose $y_t \in \mathcal{Y} := [0,1]$, and the payoff vector will be $\boldsymbol{\ell}_t(\mathbf{w}_t, y_t) = \mathbf{c}(\mathbf{w}_t, y_t) \oplus r(\mathbf{w}_t, q_t, y_t)$ [1] defined as follows:

$$\mathbf{c}(\mathbf{w}_t, y_t) := \left(\mathbf{w}_t(0)\left(y_t - \frac{0}{m}\right), \ldots, \mathbf{w}_t(m)\left(y_t - \frac{m}{m}\right)\right) \tag{5}$$

$$r(\mathbf{w}_t, q_t, y_t) := \sum_{i=0}^{m} \mathbf{w}_t(i)\left(S\left(\frac{i}{m}, y_t\right) - S(q_t, y_t)\right) \tag{6}$$

[1] $\oplus$ represents concatenation

The set we wish to approach is

$$\mathcal{S}^m_{\text{approach}} \tag{7}$$
$$= \left\{(x,z) \mid \ x \in \mathbb{R}^{m+1}, z \in \mathbb{R} \text{ s.t } ||x||_1 \leq \frac{1}{m},\ z \leq \frac{4L_S}{m^2}\right\} \tag{8}$$

In Section 1.4, we pointed out that Kuleshov and Ermon (2017)'s approach works by running $1/\varepsilon = m$ many calibration algorithms in parallel, one for each prediction interval $[i/m, (i+1)/m]$. Each calibration algorithm solves a vector-valued game with payoff vectors of dimension m. Thus, their approach can also be interpreted as using a quadratic number of dimensions (m^2) in the payoff vector while we show how to achieve the same low regret guarantee using a linear number of dimensions ($m+2$). Our main technical contribution is that the lower-dimensional problem we formulate requires a novel proof of approachability, which our work supplies, whereas in the higher-dimensional problem formulated implicitly by Kuleshov and Ermon (2017) approachability follows "for free" due to a more general result by (Blum and Mansour, 2007; Cesa-Bianchi and Lugosi, 2006).

3.2 Proof of Approachability

For the calibration vector-payoff game, Abernethy et al. (2011) prove approachability via response-satisfiability. While this is arguably the simplest way to prove approachability, it is important to note that to construct an algorithm for approaching the desired set, simply proving response-satisfiability is not enough. A halfspace oracle needs to be provided as well. Although Abernethy et al. (2011) prove approachability by response-satisfiability, they present a halfspace oracle based on the construction in Foster's halfspace-approachability proof. For our recalibration problem, we prove approachability by showing halfspace-approachability. Our proof is constructive, hence it directly yields a halfspace oracle.

Theorem 6. *For the vector-valued game defined in 5, the set $\mathcal{S} = \mathcal{S}^m_{approach}$ is approachable. That is, any halfspace H containing $\mathcal{S}$ is approachable.*

Proof. First we characterize the set of halfspaces containing S. Let H be a halfspace of $\mathbb{R}^{m+2}$ defined by the equation $\langle a, x\rangle + bz \leq \theta$ for $x \in \mathbb{R}^{m+1}, z \in \mathbb{R}$. We claim that $\mathcal{S} \subseteq \mathcal{H}$ iff $b \geq 0$ and $\theta \geq \left(\frac{||a||_\infty}{m} + \frac{4bL_S}{m^2}\right)$. To see this, observe that for $\mathcal{H}$ to contain S. It must be the case that

$$\max\left\{\langle a, x\rangle + bz \mid \ ||x||_1 \leq \frac{1}{m}, z \leq \frac{4L_S}{m^2}\right\} \leq \theta$$

First, we need $b \geq 0$, since we can choose z to violate this constraint otherwise. Secondly, we need $\theta \geq \left(\frac{||a||_\infty}{m} + \frac{4bL_S}{m^2}\right)$, since we can choose x and z to violate this constraint otherwise. Thus, if $\mathcal{S} \subseteq H$, then both conditions $b \geq 0$ and $\theta \geq \left(\frac{||a||_\infty}{m} + \frac{4bL_S}{m^2}\right)$ must hold for H. Conversely, if both conditions $b \geq 0$ and $\theta \geq \left(\frac{||a||_\infty}{m} + \frac{4bL_S}{m^2}\right)$ hold for H, then $\mathcal{S} \subseteq H$. This is because for any $(x,z) \in \mathcal{S}$, $\langle a, x\rangle + bz \leq \left(\frac{||a||_\infty}{m} + \frac{4bL_S}{m^2}\right) \leq \theta$ and if $b < 0$, we can obtain a contradiction by choosing $z < -\frac{\theta}{b}$.

WLOG, we will assume $\theta = \left(\frac{||a||_\infty}{m} + \frac{4bL_S}{m^2}\right)$, since approachability of a halfspace defined by $\langle a, x\rangle + bz \leq \left(\frac{||a||_\infty}{m} + \frac{4bL_S}{m^2}\right)$ implies approachability of $\langle a, x\rangle + bz \leq \theta$ for $\theta \geq \left(\frac{||a||_\infty}{m} + \frac{4bL_S}{m^2}\right)$. That is, we will only concern ourselves with proving halfspace-approachability for halfspaces such that $\theta = \left(\frac{||a||_\infty}{m} + \frac{4bL_S}{m^2}\right)$. For a halfspace such that $a = \mathbf{0}$, we follow the halfspace oracle in 17 and set $a_i = 0$ for all i. This gives us regret at most $\frac{4L_S}{m^2}$; see proof of 7 in the appendix. If $a \neq \mathbf{0}$, then we can consider the halfspace normalized by $||a||_\infty$, that is, the halfspace defined by $a' = \frac{a}{||a||_\infty}, b' = \frac{b}{||a||_\infty}$ and $\theta = \frac{1}{m} + \frac{4b'L_S}{m^2}$. Since $||a'||_\infty = 1$ and $b' \geq 0$, by Lemma 7, this halfspace is approachable. Consequently, any halfspace containing $\mathcal{S}$ is approachable. □

Lemma 7. *Consider a pair $(a,b) \in \mathbb{R}^{m+1} \times \mathbb{R}$ such that $\|a\|_\infty = 1$ and $b \geq 0$. The halfspace H_1, defined below, is approachable.*

$$H_1 := \left\{(x,z) \in \mathbb{R}^{m+1} \times \mathbb{R} \;\middle|\; \langle a, x\rangle + bz \leq \frac{1}{m} + \frac{4bL_S}{m^2}\right\} \tag{9}$$

The full proof can be found in the appendix. We provide a proof sketch here. To show that H_1 is approachable, we will find a mixed distribution for the forecaster (i.e, a probability distribution over $p \in \{0/m, 1/m, 2/m, \ldots, (m-1)/m, 1\}$) such that $\mathbb{E}_p\left[\langle a, \mathbf{c}(p,y)\rangle + br(p, q_t, y)\right] \leq \frac{1}{m} + \frac{4bL_S}{m^2}$ for any $y \in \{0,1\}$. For simplicity, define

$$f(i,y) = a_i\left(\frac{i}{m} - y\right) + b\left[S\left(\frac{i}{m}, y\right) - S(q_t, y)\right] \tag{10}$$

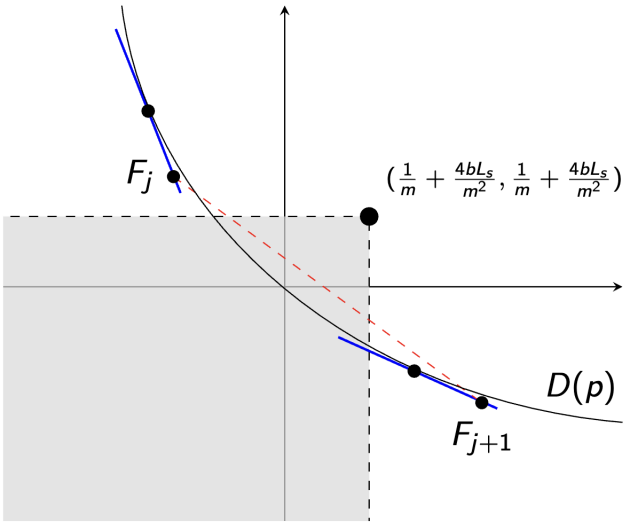
$$F_i = \begin{bmatrix} f(i,0) \\ f(i,1)\end{bmatrix} \tag{11}$$

so our objective becomes to show that there exists a distribution p over $\frac{i}{m} \in \{0, \ldots, m\}$ such that $\mathbb{E}_p f(i,y) \leq \frac{1}{m} + \frac{4bL_S}{m^2}$ for $y \in \{0,1\}$, or equivalently that the vector $\mathbb{E}_p F_i$ belongs to the quadrant-shaped set $(-\infty, \frac{1}{m} + \frac{4bL_S}{m^2}] \times (-\infty, \frac{1}{m} + \frac{4bL_S}{m^2}]$. We will be

choosing p to be either a point-mass on $\frac{i}{m}$ for some i, or a distribution on two consecutive values in the set $\{0, \frac{1}{m}, \frac{2}{m}, \ldots, 1\}$. For $p \in [0,1]$ let $D(p)$ denote the vector corresponding to the scoring rule term in F_i.

$$D(p) = b \cdot \begin{bmatrix} S(p,0) - S(q_t,0) \\ S(p,1) - S(q_t,1) \end{bmatrix}$$

As a result of the fact that S is a proper scoring rule, an important observation is that the curve formed by $D(p)$ is convex and its tangent lines are parallel to $\begin{bmatrix} p \\ p-1 \end{bmatrix}$. Thus, $F_0, F_1, \ldots, F_m$ are points on a sequence of tangent lines to the convex curve formed by $D(p)$. Additionally, we can show that F_0 lies in the left half-plane while F_m must belong to the lower half-plane. Thus, $F_0, F_1, \ldots, F_m$ are always in the second, third or fourth quadrants and lie on lines with slopes that are slowly changing from negative to positive.



If F_i belongs to the third quadrant — that is, the set $(-\infty, 0] \times (-\infty, 0]$ — then we choose p to be a point-mass on i. This guarantees that $\mathbb{E}_p F_i \leq \frac{1}{m} + \frac{4bL_S}{m^2}$. Otherwise, there must be at least one index j such that F_j lies in the second quadrant while F_{j+1} lies in the fourth quadrant. Using plane geometry, we show that the line segment joining F_j and F_{j+1} intersects the set $(-\infty, \frac{1}{m} + \frac{4bL_S}{m^2}] \times (-\infty, \frac{1}{m} + \frac{4bL_S}{m^2}]$ as required. The rest of the proof can be found in the appendix.

3.3 Efficient Algorithm via Online Linear Optimization

We now show how the results in the previous section lead to an efficient algorithm for online recalibration. The steps in this section are parallel to those in Section 5.2 of Abernethy et al. (2011) but we have to repeat them because our payoff game and convex sets are different.

Theorem 8. *For any m, there exists a $(\ell_1, \frac{1}{m}, \frac{4L_S}{m^2})$-online recalibration algorithm that runs in time $O(\log m)$ per iteration and guarantees an expected recalibration rate of $O\left(\sqrt{\frac{m}{T}}\right)$*

Following the steps of the reduction from Approachability to OLO outlined in Abernethy et al. (2011), we provide a convex set $\mathcal{K}$ whose elements correspond to halfspaces containing $\mathcal{S}^m_{\text{approach}}$, and express the distance of a loss vector to the set $\mathcal{S}$ we wish to approach as an optimization over the convex set $\mathcal{K}$. We do so in Lemma 9. Then, we present an algorithm (halfspace oracle) such that given a halfspace $\boldsymbol{\theta}_t \in \mathcal{K}$, it returns a distribution $\mathbf{w}_t \in \Delta_{m+1}$ with the guarantee that $\langle \boldsymbol{\ell}_t(\mathbf{w}_t, y_t), \boldsymbol{\theta}_t \rangle \leq \frac{1}{m} + \frac{4L_S}{m^2}$. Lastly, we present an algorithm for recalibration that uses Online Gradient Descent Zinkevich (2003) to select the halfspace $\boldsymbol{\theta}_t \in \mathcal{K}$ to approach at each timestep.

We define the convex set $\mathcal{K}$ as follows

$$\left\{(a,b) \mid \ a \in \mathbb{R}^{m+1}, b \in \mathbb{R} \text{ s.t } ||a||_\infty \leq 1, \ 0 \leq b \leq 1\right\} \tag{12}$$

This is an appropriate choice of $\mathcal{K}$ due to Lemma 9, since it allows us to upper bound the distance to S in terms of a linear optimization objective over the set $\mathcal{K}$.

Lemma 9. *For any vector $\mathbf{x} \in \mathbb{R}^{m+2}$ such that $\|\mathbf{x}_{1:m+1}\|_1 \geq 1/m$, and $|\mathbf{x}_{m+2}| \geq \frac{4L_S}{m^2}$,*

$$\texttt{dist}_1\left(\mathbf{x}, \mathcal{S}^m_{approach}\right) = -\frac{1}{m} - \frac{4L_S}{m^2} - \min_{\theta \in \mathcal{K}} \langle -\mathbf{x}, \theta \rangle \tag{13}$$

We defer the proof of Lemma 9 to the appendix. The usefulness of the lemma above is that it allows us to combine the approachability guarantee of Theorem 6 to upper bound the distance to the target convex set in terms of regret of an online linear optimization algorithm.

$$\texttt{dist}_1\left(\frac{1}{T}\sum_{t=1}^{T} \boldsymbol{\ell}_t(\mathbf{w}_t, y_t), \mathcal{S}^m_{\text{approach}}\right) \tag{14}$$

$$= -\frac{1}{m} - \frac{4L_S}{m^2} - \min_{\theta \in \mathcal{K}} \left\langle -\frac{1}{T}\sum_{t=1}^{T} \boldsymbol{\ell}_t(w_t, y_t), \boldsymbol{\theta} \right\rangle \tag{15}$$

$$\leq \frac{1}{T}\left(\sum_{t=1}^{T} \langle -\boldsymbol{\ell}_t(\mathbf{w}_t, y_t), \boldsymbol{\theta}_t \rangle - \min_{\boldsymbol{\theta} \in \mathcal{K}} \sum_{t=1}^{T} \langle -\boldsymbol{\ell}_t(\mathbf{w}_t, y_t), \boldsymbol{\theta} \rangle\right) \tag{16}$$

where the inequality follows from the approachability guarantee of Theorem 6: for any halfspace θ_t, there exists a distribution $\mathbf{w}_t$ such that $\langle \boldsymbol{\ell}_t(\mathbf{w}_t, y_t), \theta_t \rangle \leq \frac{1}{m} + \frac{4L_S}{m^2}$ for any $y_t \in \{0, 1\}$.

The Halfspace Oracle: Approach($\boldsymbol{\theta}_{t+1}$) Given any $\boldsymbol{\theta}_t \in \mathcal{K}$, we must construct $\mathbf{w} \in \Delta_{m+1}$ so that

$\langle \boldsymbol{\ell}_t(\mathbf{w}_t, y_t), \boldsymbol{\theta}_t \rangle \leq \frac{1}{m} + \frac{4L_S}{m^2}$ for any y_t. The proof of approachability for Lemma 7 is a constructive one and describes how to choose $\mathbf{w}_t \in \Delta_{m+1}$ given $\boldsymbol{\theta}_t$. Recall functions $f(i, y)$ and F_i defined in 10. The algorithm firsts check if F_0 or F_m is in the 3rd quadrant. If one of them is, then we output a point distribution at the corresponding probability value. If none of F_0 or F_m is in the 3rd quadrant, then we binary search for an index i with F_i in the 3rd quadrant or a pair of consecutive indices $j, j+1$ where F_j is in 2nd quadrant and F_{j+1} is in the 4th quadrant. In the first case, $wv_t(i) = 1$ and 0 everywhere else. In the second case, we set

$$\mathbf{w}_t(j) = \frac{f(j+1,1) - f(j+1,0)}{f(j,0) - f(j+1,0) - f(j,1) + f(j+1,1)} \tag{17}$$

$$\mathbf{w}_t(j+1) = \frac{f(j,0) - f(j,1)}{f(j,0) - f(j+1,0) - f(j,1) + f(j+1,1)} \tag{18}$$

and 0 everywhere else. The correctness of this procedure follows from the proof of Lemma 7. Note that F_i does not need to be pre-computed for every index. It can be computed online during the binary search steps. Thus, this halfspace oracle can be implemented in $O(\log m)$ steps.

The Learning Algorithm: OGD($\boldsymbol{\theta}_t | l_t$) Similar to Abernethy et al. (2011), we use the Online Gradient Descent algorithm (Zinkevich, 2003) as the learning algorithm.

Algorithm 1 Online Recalibration Algorithm

Input: some natural number $m \geq \sqrt{4L_S}$
Initialize: $\boldsymbol{\theta}_1 = \mathbf{0}, \mathbf{w}_1 \in \Delta_{m+1}$
for $t = 1, \ldots, T$ **do**
  Observe q_t from black-box prediction oracle
  Sample $i_t \sim \mathbf{w}_t$, predict $p_t = \frac{i_t}{m}$, observe y_t
  Set $l_t := -\boldsymbol{\ell}_t(\mathbf{w}_t, y_t)$
  Query learning algorithm: $\boldsymbol{\theta}_{t+1} \leftarrow \text{OGD}(\boldsymbol{\theta}_t | l_t)$
  // Online Gradient Descent step
  Query halfspace oracle: $\mathbf{w}_{t+1} \leftarrow \text{Approach}(\boldsymbol{\theta}_{t+1})$
  // Obtain $\mathbf{w}_{t+1} \in \Delta_{m+1}$ from $\boldsymbol{\theta}_{t+1}$
end for

OGD guarantees that the regret is no more than $DG\sqrt{T}$ where D is the ℓ_2 diameter of the set and G is the ℓ_2-norm of the largest cost vector. For the convex set $\mathcal{K}$, the ℓ_2 diameter is $O(\sqrt{m})$. The ℓ_2-norm of the calibration component of the vector is bounded by $\sqrt{2}$. To make the size of the regret at time t small and at most 1, we normalize by the lipschitz-constant L_S

$$C_T^{\varepsilon,\delta}(\mathcal{A}) \leq \texttt{dist}_1 \left(\frac{1}{T} \sum_{t=1}^{T} \boldsymbol{\ell}_t(\mathbf{w}_t, y_t), \mathcal{S}^m_{\text{approach}} \right) \tag{19}$$

$$\leq \frac{\text{Regret}_t}{T} \leq \frac{GD}{\sqrt{T}} = O\left(\sqrt{\frac{m}{T}}\right) \tag{20}$$

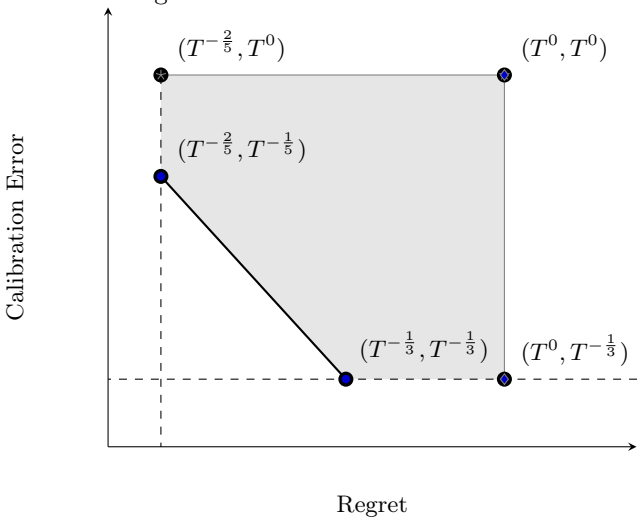
4 Convergence Rates

In this section, we describe how the results from the previous sections can be used to obtain bounds on calibration error and regret.

Theorem 10. *For any $x \in [\frac{1}{3}, \frac{2}{5}]$, given a black-box prediction oracle, there exists a forecasting algorithm that simultaneously achieves expected regret $O(T^{-x})$ while keeping the expected ℓ_1-calibration error less than T^{2x-1}.*

Proof. In Theorem 8, we show that for any m, there exists an $(\ell_1, \frac{1}{m}, \frac{4L_s}{m^2})$-online recalibration algorithm which satisfies a recalibration rate of $O\left(\sqrt{\frac{m}{T}}\right)$. By definition 4, this implies that the ℓ_1-calibration error is upper bounded by $O(\frac{1}{m} + \sqrt{\frac{m}{T}})$ and the regret is upper bounded by $O(\frac{1}{m^2} + \sqrt{\frac{m}{T}})$. Setting $m = \lceil T^{1-2x} \rceil$, we obtain an algorithm that guarantees expected regret of $O(T^{-x})$ and expected calibration error $O(T^{2x-1})$ □

Figure 1: The graph below captures the linear tradeoff between regret and ℓ_1 calibration error. According to Theorem 10, the set of jointly achievable rates contains the shaded region.



References

Abernethy, J., Bartlett, P. L., and Hazan, E. (2011). Blackwell approachability and no-regret learning are

equivalent. In Kakade, S. M. and von Luxburg, U., editors, *Proceedings of the 24th Annual Conference on Learning Theory*, volume 19 of *Proceedings of Machine Learning Research*, pages 27–46, Budapest, Hungary. PMLR.

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

Blum, A. and Mansour, Y. (2007). From external to internal regret. *Journal of Machine Learning Research*, 8(47):1307–1324.

Bolukbasi, T., Chang, K.-W., Zou, J., Saligrama, V., and Kalai, A. (2016). Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, page 4356–4364, Red Hook, NY, USA. Curran Associates Inc.

Buolamwini, J. and Gebru, T. (2018). Gender shades: Intersectional accuracy disparities in commercial gender classification. In Friedler, S. A. and Wilson, C., editors, *Proceedings of the 1st Conference on Fairness, Accountability and Transparency*, volume 81 of *Proceedings of Machine Learning Research*, pages 77–91. PMLR.

Cesa-Bianchi, N. and Lugosi, G. (2006). *Prediction, learning, and games*. Cambridge university press.

DeGroot, M. H. and Fienberg, S. E. (1983). The comparison and evaluation of forecasters. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 32(1-2):12–22.

Foster, D. P. (1999). A proof of calibration via blackwell's approachability theorem. *Games and Economic Behavior*, 29(1):73–78.

Foster, D. P. and Hart, S. (2021). Calibeating: Beating forecasters at their own game.

Gneiting, T. and Raftery, A. E. (2007). Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378.

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. (2017). On calibration of modern neural networks. In Precup, D. and Teh, Y. W., editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1321–1330. PMLR.

Hebert-Johnson, U., Kim, M., Reingold, O., and Rothblum, G. (2018). Multicalibration: Calibration for the (Computationally-identifiable) masses. In Dy, J. and Krause, A., editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1939–1948. PMLR.

Jiang, X., Osl, M., Kim, J., and Ohno-Machado, L. (2011). Calibrating predictive model estimates to support personalized medicine. *Journal of the American Medical Informatics Association*, 19(2):263–274.

Kuleshov, V. and Ermon, S. (2017). Estimating uncertainty online against an adversary. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, page 2110–2116. AAAI Press.

Lee, D., Noarov, G., Pai, M., and Roth, A. (2022). Online minimax multiobjective optimization: Multicalibeating and other applications. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A., editors, *Advances in Neural Information Processing Systems*, volume 35, pages 29051–29063. Curran Associates, Inc.

Mannor, S. and Stoltz, G. (2010). A geometric proof of calibration. *Math. Oper. Res.*, 35(4):721–727.

Niculescu-Mizil, A. and Caruana, R. (2005). Predicting good probabilities with supervised learning. In *Proceedings of the 22nd International Conference on Machine Learning*, ICML '05, page 625–632, New York, NY, USA. Association for Computing Machinery.

Platt, J. C. (1999). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *ADVANCES IN LARGE MARGIN CLASSIFIERS*, pages 61–74. MIT Press.

Zinkevich, M. (2003). Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the Twentieth International Conference on International Conference on Machine Learning*, ICML'03, page 928–935. AAAI Press.

Checklist

The checklist follows the references. For each question, choose your answer from the three possible options: Yes, No, Not Applicable. You are encouraged to include a justification to your answer, either by referencing the appropriate section of your paper or providing a brief inline description (1-2 sentences). Please do not modify the questions. Note that the Checklist section does not count towards the page limit. Not including the checklist in the first submission won't result in desk rejection, although in such case we will ask you to upload it during the author response period and include it in camera ready (if accepted).

In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [**Yes**/No/Not Applicable]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [**Yes**/No/Not Applicable]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes/No/**Not Applicable**]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [**Yes**/No/Not Applicable]
   (b) Complete proofs of all theoretical results. [**Yes**/No/Not Applicable]
   (c) Clear explanations of any assumptions. [**Yes**/No/Not Applicable]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes/No/Not **Applicable**]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes/No/**Not Applicable**]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes/No/**Not Applicable**]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes/No/**Not Applicable**]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Yes/No/**Not Applicable**]
   (b) The license information of the assets, if applicable. [Yes/No/**Not Applicable**]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Yes/No//**Not Applicable**]
   (d) Information about consent from data providers/curators. [Yes/No/**Not Applicable**]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Yes/No/**Not Applicable**]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Yes/No/**Not Applicable**]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Yes/No/**Not Applicable**]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Yes/No/**Not Applicable**]

A Appendix

Lemma 11. *For every $q_t \in [0,1]$ there exists a $\mathbf{w} \in \Delta_{m+1}$ such that for all $y \in \{0,1\}$, $r_t(\mathbf{w}, y) \leq \frac{2L_s}{m^2}$.* [2]

Proof. Fix $q = q_t$. Recalling the definition of $r_t(\mathbf{w}, y)$ in Equation (6), we see that the lemma is equivalent to proving

$$\min_{\mathbf{w} \in \Delta_{m+1}} \max_{y \in \{0,1\}} \sum_{i=0}^{m} \mathbf{w}_t(i) \left(S\left(\frac{i}{m}, y\right) - S(q, y) \right) \leq \frac{2L_s}{m^2}. \tag{21}$$

The functions $S\left(\frac{i}{m}, y\right)$ and $S(q, y)$ appearing on the right side of (21) are affine functions of y, so we can enlarge the domain of y to be the compact, convex set $[0,1]$, rather than the two-element set $\{0,1\}$, and then apply von Neumann's Minimax Theorem to conclude that inequality (21) is equivalent to

$$\max_{y \in [0,1]} \min_{\mathbf{w} \in \Delta_{m+1}} \sum_{i=0}^{m} \mathbf{w}(i) \left(S\left(\frac{i}{m}, y\right) - S(q, y) \right) \leq \frac{2L_s}{m^2}. \tag{22}$$

The inequality (22) is easy to prove. For any $y \in [0,1]$, choose $k \in [m]$ such that $|y - \frac{k}{m}| \leq \frac{1}{m}$, and let $x = \frac{k}{m}$. By the Lipschitz property of S we have

$$|S(x,0) - S(y,0)| \leq L_s |x - y| \leq \frac{L_s}{m} \quad \text{and} \quad |S(y,1) - S(x,1)| \leq L_s |x - y| \leq \frac{L_s}{m},$$

so the triangle inequality implies

$$|S(x,0) - S(y,0) + S(y,1) - S(x,1)| \leq \frac{2L_s}{m},$$

and hence

$$(x - y)[S(x,0) - S(y,0) + S(y,1) - S(x,1)] \leq \frac{1}{m} \cdot \frac{2L_s}{m} = \frac{2L_s}{m^2}. \tag{23}$$

Now, using the fact that S is a strictly proper scoring rule we have

$$\begin{aligned} S(x,y) - S(q,y) &\leq S(x,y) - S(q,y) + [S(q,y) - S(y,y)] + [S(y,x) - S(x,x)] \\ &= [S(x,y) - S(y,y)] + [S(y,x) - S(x,x)] \\ &= (1-y)[S(x,0) - S(y,0)] + y[S(x,1) - S(y,1)] \\ &\quad + (1-x)[S(y,0) - S(x,0)] + x[S(y,1) - S(x,1)] \\ &= (x-y)[S(x,0) - S(y,0) + S(y,1) - S(x,1)] \leq \frac{2L_s}{m^2}. \end{aligned}$$

Therefore, if we set $\mathbf{w}$ to be the probability vector defined by $\mathbf{w}(k) = 1$ and $\mathbf{w}(j) = 0$ for all $j \neq k$, we have

$$\sum_{i=0}^{m} \mathbf{w}(i) \left(S\left(\frac{i}{m}, y\right) - S(q,y) \right) = \left(S\left(\frac{k}{m}, y\right) - S(q,y) \right) = S(x,y) - S(q,y) \leq \frac{2L_s}{m^2}.$$

As $y \in [0,1]$ was arbitrary, we have shown that inequality (22) holds, completing the proof of the lemma. □

A.1 Proof of Lemma 7

Lemma 12. *Consider a pair $(a, b) \in \mathbb{R}^{m+1} \times \mathbb{R}$ such that $\|a\|_\infty = 1$ and $b \geq 0$. The halfspace H_1, defined below, is approachable.*

$$H_1 := \left\{ (x, z) \in \mathbb{R}^{m+1} \times \mathbb{R} \;\middle|\; \langle a, x \rangle + bz \leq \frac{1}{m} + \frac{4bL_S}{m^2} \right\} \tag{9}$$

[2] $r_t(\mathbf{w}, y)$ should be interpreted as $r_t(\mathbf{w}, q_t, y)$

Proof. To show that H_1 is approachable, we will find a mixed distribution for the forecaster (i.e, a probability distribution over $p \in \{0/m, 1/m, 2/m, \ldots, (m-1)/m, 1\}$) such that $\mathbb{E}_p\left[\langle a, \boldsymbol{\ell}_c(p,y)\rangle + b\boldsymbol{\ell}_r(p,y)\right] \le \frac{1}{m} + \frac{4bL_S}{m^2}$ for any $y \in \{0,1\}$. For simplicity, define

$$c(i,y) = \frac{i}{m} - y \quad \text{and} \quad C_i = \begin{bmatrix} c(i,0) \\ c(i,1) \end{bmatrix}$$
$$d(i,y) = b \cdot S\left(\frac{i}{m}, y\right) - b \cdot S(q_t, y) \quad \text{and} \quad D_i = \begin{bmatrix} d(i,0) \\ d(i,1) \end{bmatrix}$$
$$f(i,y) = a_i c(i,y) + d(i,y) \quad \text{and} \quad F_i = \begin{bmatrix} f(i,0) \\ f(i,1) \end{bmatrix} = a_i C_i + D_i$$

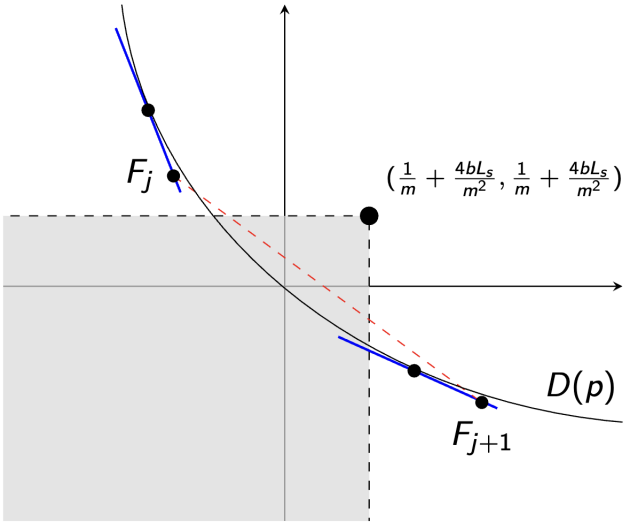
Observe that $f(i,y) = \langle a, \boldsymbol{\ell}_c(\frac{i}{m}, y)\rangle + b\boldsymbol{\ell}_r(\frac{i}{m}, y)$, so our objective becomes to show that there exists a distribution p over $\frac{i}{m} \in \{0, \ldots, m\}$ such that $\mathbb{E}_p\, f(i,y) \le \frac{1}{m} + \frac{4bL_S}{m^2}$ for $y \in \{0,1\}$, or equivalently that the vector $\mathbb{E}_p\, F_i$ belongs to the quadrant-shaped set $(-\infty, \frac{1}{m} + \frac{4bL_S}{m^2}] \times (-\infty, \frac{1}{m} + \frac{4bL_S}{m^2}]$. We will be choosing p to be either a point-mass on $\frac{i}{m}$ for some i, or a distribution on two consecutive values in the set $\{0, \frac{1}{m}, \frac{2}{m}, \ldots, 1\}$. Hence, the vector $\mathbb{E}_p\, F_i$ will belong to one of m closed line segments forming a polygonal path through the vectors $F_0, F_1, \ldots, F_m$. Observe that F_0 belongs to the left half-plane, i.e. $f(0,0) \le 0$, because

$$f(0,0) = a_0 c(0,0) + d(0,0) = b(S(0,0) - S(q_t, 0)) \le 0,$$

where the last inequality holds because $b \ge 0$ and S is a proper scoring rule. Similarly, F_m belongs to the lower half-plane, i.e. $f(m,1) \le 0$, because

$$f(m,1) = a_m c(m,1) + d(m,1) = b(S(1,1) - S(q_t, 1)) \le 0.$$

If F_0 or, respectively, F_m belongs to the third quadrant — that is, the set $(-\infty, 0] \times (-\infty, 0]$ — then we choose p to be a point-mass on 0 or 1, respectively. The remaining case is that F_0 and F_m belong to the sets $(-\infty, 0] \times (0, \infty)$ and $(0, \infty) \times (-\infty, 0]$, respectively. In that case, F_0 and F_m lie on opposite sides of the line L consisting of all points $\begin{bmatrix} x_0 \\ x_1 \end{bmatrix}$ that satisfy $x_0 = x_1$; F_0 lies above L while F_m lies below it. Hence, there must be at least one index j such that F_j lies on or above L while F_{j+1} lies below it. We aim to construct a distribution p supported on $\{\frac{j}{m}, \frac{j+1}{m}\}$ such that $\mathbb{E}_p\, F_i$ belongs to the set $(-\infty, \frac{1}{m} + \frac{4bL_S}{m^2}] \times (-\infty, \frac{1}{m} + \frac{4bL_S}{m^2}]$. Assume without loss of generality that $j \ge m/2$. (The case $j \le m/2$ is handled symmetrically, by exchanging the roles of the labels $y = 0$ and $y = 1$, i.e. the first and second coordinates of the vectors we are considering.)



For $p \in [0,1]$ let $D(p)$ denote the vector

$$D(p) = b \cdot \begin{bmatrix} S(p,0) - S(q_t,0) \\ S(p,1) - S(q_t,1) \end{bmatrix}$$

and observe that the notation D_i defined earlier is equivalent to $D(i/m)$. The fact that S is a proper scoring rule ensures that when y is a random sample from $\{0,1\}$ taking the value 1 with some probability p, the value of p' that minimizes $\mathbb{E}_y[S(p',y) - S(q_t,y)]$ is $p' = p$. Since the expected value $\mathbb{E}_y[S(p',y) - S(q_t,y)]$ is calculated by taking the inner product of the vector $D(p')$ with the probability vector

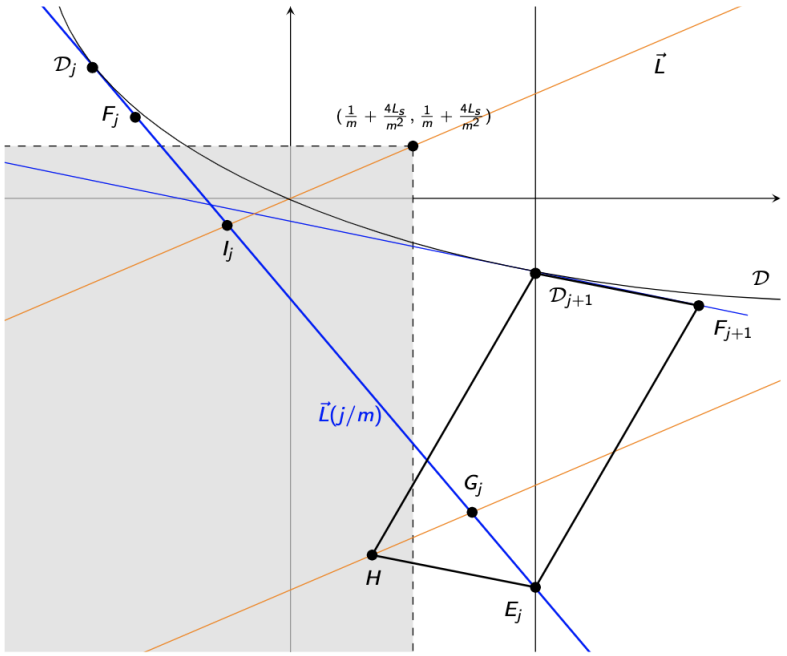
$$Y(p) = \begin{bmatrix} 1-p \\ p \end{bmatrix},$$

this means that the curve $\mathcal{D} = \{D(p') \mid 0 \le p' \le 1\}$ is convex and that the line

$$L(p) = \{x \mid \langle Y(p), x\rangle = \langle Y(p), D(p)\rangle\}$$

is tangent to $\mathcal{D}$ at the point $D(p)$. The normal vector to this tangent line is $Y(p)$, so the vector $C(p) = \begin{bmatrix} p \\ p-1 \end{bmatrix}$, being orthogonal to $Y(p)$, is parallel to the tangent line at $D(p)$. When $p = i/m$, observe that the vector $C(p)$ defined here coincides with C_i defined earlier.

Summarizing the foregoing discussion, the line $L(j/m) = \{D_j + \lambda C_j \mid \lambda \in \mathbb{R}\}$ is tangent to the convex curve $\mathcal{D}$ at the point D_j, hence it lies (weakly) below that curve. In particular, recall the line L consisting of points whose first and second coordinates are equal, and consider the point I_j where L intersects $L(j/m)$. Since $L(j/m)$ lies (weakly) below $\mathcal{D}$ and $\mathcal{D}$ intersects L at $D(q_t) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, the intersection of $L(j/m)$ with L must belong to the third quadrant. From these properties, it will follow that the line segment joining F_j to F_{j+1} intersects the set $(-\infty, \frac{1}{m} + \frac{4bL_S}{m^2}] \times (-\infty, \frac{1}{m} + \frac{4bL_S}{m^2}]$ as required.



Let E_j be the intersection point of $L(j/m)$ with a vertical line through D_{j+1}. Since $L(j/m)$ lies below $\mathcal{D}$, we know that E_j is situated directly below D_{j+1}. To reason about the distance between D_{j+1} and E_j, observe that the convexity of the curve $\mathcal{D}$ implies that the slope of the line segment joining D_j to D_{j+1} lies between the slopes of the tangent lines at D_j and D_{j+1}. Those slopes are $1 - m/j$ and $1 - m/(j+1)$, respectively. Hence, a pair of lines passing through D_j, with slopes $1 - m/j$ and $1 - m/(j+1)$, will intersect the vertical line through D_{j+1} in a line segment that contains D_{j+1}. The lower endpoint of that line segment is E_j. Its length is the difference between the slopes of the two lines, times the horizontal displacement between D_j and D_{j+1}. In other words, the length of the vertical line segment is

$$\frac{m}{j(j+1)} \cdot b \cdot \left[S\left(\frac{j}{m}, 0\right) - S\left(\frac{j+1}{m}, 0\right)\right] \le \frac{m}{(m/2)^2} \cdot \frac{bL_S}{m} = \frac{4bL_S}{m^2}.$$

Since the vertical line segment contains E_j and D_{j+1}, its length is an upper bound on their distance from one another.

Now define

$$G_j = E_j + a_{j+1}C_j = F_{j+1} + (E_j - D_{j+1}) + a_{j+1}(C_j - C_{j+1}).$$

Since E_j lies on $L(j/m)$ and C_j is parallel to $L(j/m)$, we know that G_j lies on $L(j/m)$. To determine the position of G_j relative to L, observe that $C_j - C_{j+1} = \begin{bmatrix} -1/m \\ -1/m \end{bmatrix}$ is parallel to L, $F_{j+1} + (E_j - D_{j+1})$ lies below F_{j+1}, and recall that F_{j+1} lies below L. Hence, G_j lies below L. As F_j lies on or above L it follows that the line segment joining F_j to G_j intersects L, and this intersection point must be I_j because the segment connecting F_j to G_j is contained in $L(j/m)$. Write $I_j = (1-t)F_j + tG_j$ for some parameter $t \in [0,1]$.

If p is the distribution that selects a random $\frac{i}{m} \in \{\frac{j}{m}, \frac{j+1}{m}\}$ by setting $\frac{i}{m} = \frac{j}{m}$ with probability $1-t$ and $\frac{i}{m} = \frac{j+1}{m}$ with probability t, then

$$\begin{aligned} \mathbb{E}_p F_i = (1-t)F_j + tF_{j+1} &= (1-t)F_j + tG_j + t(F_{j+1} - G_j) \\ &= I_j + t(D_{j+1} - E_j) + ta_{j+1}(C_{j+1} - C_j) \\ &= I_j + t(D_{j+1} - E_j) + \tfrac{ta_{j+1}}{m}\begin{bmatrix} 1 \\ 1 \end{bmatrix}. \end{aligned} \tag{24}$$

We need to show that both coordinates of the vector in Equation (24) are less than or equal to $\frac{1}{m} + \frac{4bL_S}{m}$. The first coordinate of I_j is non-positive, the first coordinate of $D_{j+1} - E_j$ is zero, and the first coordinate of $\frac{ta_{j+1}}{m}\begin{bmatrix} 1 \\ 1 \end{bmatrix}$ is at most $\frac{1}{m}$ since $0 \le t \le 1$ and $|a_{j+1}| \le 1$. The second coordinate of I_j is non-positive, the second coordinate of $D_{j+1} - E_j$ is at most $\frac{4bL_S}{m^2}$, and the second coordinate of $\frac{ta_{j+1}}{m}\begin{bmatrix} 1 \\ 1 \end{bmatrix}$ is at most $\frac{1}{m}$. □

A.2 Constructing the Halfspace Oracle

Here we go into more detail about how to construct the oracle asserted in Section 3.3. Recall that in the proof of Lemma A.1, given a halfspace θ parameterized by (a, b), we defined the vector F_i as follows:

$$F_i = \begin{bmatrix} f(i,0) \\ f(i,1) \end{bmatrix} \quad \text{where} \quad f(i,y) = a_i\left(\frac{i}{m} - y\right) + b\left[S\left(\frac{i}{m}, y\right) - S(q_t, y)\right] \tag{25}$$

In the proof, we note that F_0 is either in the 2nd or 3rd quadrant. Similarly, F_m is either in the 3rd or 4th quadrant. Thus, we first check if F_0 or F_m is in the 3rd quadrant. If one of them is, then we output a point distribution at the corresponding probability value. If none of F_0 or F_m is in the 3rd quadrant, then we binary search for an index i with F_i in the 3rd quadrant or a pair of consecutive indices $j, j+1$ where F_j is in 2nd quadrant and F_{j+1} is in the 4th quadrant. In the first case, $w_t(i) = 1$ and 0 everywhere else. In the second case, we set

$$w_t(j) = \frac{f(j+1,1) - f(j+1,0)}{f(j,0) - f(j+1,0) - f(j,1) + f(j+1,1)} \tag{26}$$

$$w_t(j+1) = \frac{f(j,0) - f(j,1)}{f(j,0) - f(j+1,0) - f(j,1) + f(j+1,1)} \tag{27}$$

and 0 everywhere else. The correctness of this procedure follows from the proof of Lemma A.1. The formula is obtained by solving this system of equations below to obtain a convex combination of F_j and F_{j+1}:

$$\begin{bmatrix} 1 & 1 \end{bmatrix}\begin{bmatrix} w_t(j) \\ w_t(j+1) \end{bmatrix} = 1 \quad \text{and} \quad \begin{bmatrix} 1 & -1 \end{bmatrix}\begin{bmatrix} f(j,0) & f(j+1,0) \\ f(j,1) & f(j+1,1) \end{bmatrix}\begin{bmatrix} w_t(j) \\ w_t(j+1) \end{bmatrix} = 0$$

Note that F_i does not need to be pre-computed for every index. It can be computed online during the binary search steps. Thus, this halfspace oracle can be implemented in $O(\log m)$ steps.

A.3 Proof of Lemma 9

Lemma 13. *For any vector* $\mathbf{x} \in \mathbb{R}^{m+2}$ *such that* $\|\mathbf{x}_{1:m+1}\|_1 \geq 1/m$, *and* $|\mathbf{x}_{m+2}| \geq \frac{4L_S}{m^2}$,

$$\texttt{dist}_1\left(\mathbf{x}, \mathcal{S}^m_{approach}\right) = -\frac{1}{m} - \frac{4L_S}{m^2} - \min_{\theta \in \mathcal{K}} \langle -\mathbf{x}, \theta \rangle \tag{13}$$

Proof.

$$\texttt{dist}_1\left(\mathbf{x}, \mathcal{S}^m_{\text{approach}}\right) = \texttt{dist}_1\left(\mathbf{x}_{1:m+1}, B_1^{m+1}(1/m)\right) + \texttt{dist}_1\left(\mathbf{x}_{m+2}, \left(-\infty, \frac{4L_s}{m^2}\right]\right) \tag{28}$$

$$= -\frac{1}{m} - \min_{\theta : ||\theta||_\infty \leq 1} \langle -\mathbf{x}_{1:m+1}, \theta \rangle - \frac{4L_s}{m^2} - \min_{\theta \in [0,1]} \langle -\mathbf{x}_{m+2}, \theta \rangle \tag{29}$$

$$= -\frac{1}{m} - \frac{4L_s}{m^2} - \min_{\theta \in \mathcal{K}} \langle -\mathbf{x}, \theta \rangle \tag{30}$$

$$\tag{31}$$

Remark: We need $\|\mathbf{x}_{1:m+1}\|_1 \geq 1/m$, and $|\mathbf{x}_{m+2}| \geq \frac{L_s}{m^2}$ mainly for technicality in order to ensure equality. If these didn't hold, just like in the proof of Approachability, if you're already in the set you wish to approach, you can just make an arbitrary move. Similarly, if $\|\mathbf{x}_{1:m+1}\|_1 < 1/m$ (i.e calibration error is already less than $\frac{1}{m}$), the algorithm can just follow the oracle's predictions. On the other hand, if $\mathbf{x}_{m+2} < \frac{L_s}{m^2}$, then following the halfspace oracle still ensures expected calibration error of at most $\frac{1}{m}$ for the timestep. □

B Reducing Recalibration to Online Multiobjective Optimization

In this section we present a reduction from recalibration to the online multiobjective optimization problem studied by Lee et al. (2022). We begin by reviewing their assumptions and terminology and restating their main result.

B.1 Review of Online Multiobjective Optimization

In the setting considered by Lee et al. (2022), a learner and an adversary play a T-round game where the timing and information structure of each round t are as follows.

1. The adversary presents to the learner an *environment* (X^t, Y^t, ℓ^t) where each of X^t, Y^t is a compact convex subset of a finite-dimensional Euclidean space, and ℓ^t is a continuous vector-valued loss function taking values in $[-C, C]^d$, such that each coordinate function $\ell^t_j(x, y)\,(j = 1, 2, \ldots, d)$ is convex in its first argument and concave in its second argument.
2. The learner chooses $x^t \in X^t$ and reveals it to the adversary.
3. The adversary chooses $y^t \in Y^t$ and reveals it to the learner.

The game ends after T rounds, and the cumulative loss vector is $\sum_{t=1}^{T} \ell^t(x^t, y^t)$. The learner's objective is to minimize the maximum coordinate of this vector.

The algorithm analyzed by Lee et al. (2022) is easy to describe. For a specified learning rate $\eta > 0$, the algorithm computes in each round a weight vector χ^t whose j^{th} coordinate is proportional to $\exp(\eta \sum_{s=1}^{t-1} \ell^s_j(x^s, y^s))$. Then it chooses x^t by solving the minimax problem

$$x^t \in \arg\min_{x \in X^t} \max_{y \in Y^t} \langle \chi^t,\, \ell^t(x, y) \rangle.$$

The analysis of the algorithm relates the learner's loss to a quantity called the *AMF value*, defined as follows. For the environment (X^t, Y^t, ℓ^t) selected by the adversary at time t, define w^t_A by

$$w^t_A = \sup_{y \in Y^t} \min_{x \in X^t} \left\{ \max_{j \in [d]} \ell^t_j(x, y) \right\}.$$

This quantity w^t_A is called the *AMF value* of the stage-t environment (X^t, Y^t, ℓ^t) because it is the value of the game in which the adversary moves first, announcing $y \in Y^t$, the learner responds by selecting $x \in X^t$, and the learner seeks to minimize the loss function $\max_{j \in [d]} \ell^t_j(x, y)$. (The abbreviation "AMF" stands for "adversary moves first".)

Theorem 14 (Lee et al. (2022))**.** *Suppose $T \geq \ln(d)$. If the learner uses the multiplicative-weights algorithm described above, with learning rate $\eta = \sqrt{\frac{\ln d}{4TC^2}}$, then its cumulative loss vector will satisfy*

$$\max_{j \in [d]} \sum_{t=1}^{T} \ell^t_j(x^t, y^t) \leq \sum_{t=1}^{T} w^t_A \;+\; 4C\sqrt{T \ln d}. \tag{32}$$

B.2 Reducing Recalibration to Online Multiobjective Optimization

Suppose we are given $\varepsilon = 1/m$ for some natural number m, and we wish to design a recalibration algorithm that predicts probabilities p_t in the set $\{0,\, 1/m,\, 2/m,\, \ldots, 1\}$. Recall the vector-payoff game from Section 3 that was used for recalibration. Adjusting notation to match the notation from Lee et al. (2022), the forecasting algorithm uses a distribution x^t drawn from $X^t = \Delta_{m+1}$, the set of probability distributions on the $(m+1)$-element set $\{0, 1/m, \ldots, 1\}$. (In Section 3 this distribution was called $\mathbf{w}_t$.) The adversary selects y^t from $Y^t = [0, 1]$. (Formerly this was called y_t and constrained to belong to $\{0, 1\}$.) The vector payoff $\ell^t(x, y)$ is defined to be $\ell^t(x, y) = \mathbf{c}(x, y) \oplus r_t(x, y)$, where

$$\mathbf{c}_i(x, y) = x_i \left(y - \frac{i}{m} \right) \tag{33}$$

$$r_t(x, y) = \sum_{i=0}^{m} x_i \left(S\left(\frac{i}{m}, y \right) - S(q_t, y) \right). \tag{34}$$

After T rounds of interaction, if we write the average loss vector $\bar{\ell} = \frac{1}{T}\sum_{t=1}^{T} \ell^t(x^t, y^t)$ as

$$\bar{\ell} = \left(\frac{1}{T}\sum_{t=1}^{T} \mathbf{c}(x^t, y^t)\right) \oplus \left(\frac{1}{T}\sum_{t=1}^{T} r_t(x^t, y^t)\right) = \bar{\mathbf{c}} \oplus \bar{r}$$

then ℓ_1 calibration error is $\|\bar{\mathbf{c}}\|_1$ while the average regret is $\bar{r}$.

The objective in the recalibration problem is to ensure that $\|\bar{\mathbf{c}}\|_1$ and $\bar{r}$ are both small. This doesn't quite correspond to the learner's goal in online multiobjective optimization, which is to make every *coordinate* of the average loss vector small. The difference is that in recalibration we are concerned with $\|\bar{\mathbf{c}}\|_1$ rather than $\|\bar{\mathbf{c}}\|_\infty$. However, the difference can be overcome by embedding the loss vectors in a higher dimension. Specifically, let $d = 2^{m+1}+1$ and let M be the matrix with d rows and $m+2$ columns such that the first $d-1$ rows of M constitute the set of row vectors $\{\pm 1\}^{m+1} \oplus (0)$ while the last row of M is the row vector $(0)^{m+1} \oplus (1) = (0, 0, \ldots, 0, 1)$. For any vector $w = c \oplus r \in \mathbb{R}^{m+1} \oplus \mathbb{R}$ we have

$$\max_{j\in[d]} (Mw)_j = \max\{\|c\|_1,\, r\}. \tag{35}$$

Hence, in the online multiobjective optimization problem with d-dimensional vector losses $\tilde{\ell}^t = M\ell^t$, the maximum coordinate of the (normalized) cumulative loss vector $\frac{1}{T}\sum_{t=1}^{T} \tilde{\ell}^t(x^t, y^t)$ equals the maximum of the forecaster's ℓ_1 calibration error and average regret.

To apply Theorem 14 to the sequence of environments $(X^t, Y^t, \tilde{\ell}^t)$ we first need upper bounds on the infinity-norms of the loss vectors $\tilde{\ell}^t(x, y)$ and on the AMF values, w_A^t, of these environments. Such upper bounds are very easy to obtain. We have

$$\|\tilde{\ell}^t(x, y)\|_\infty = \max\{\|\mathbf{c}(x, y)\|_1,\, |r_t(x, t)|\} \le \max\{1, L_s\},$$

where the inequality follows from the definitions of $\mathbf{c}(x, y)$ and $r_t(x, y)$, recalling that the Lipschitz constant of the scoring rule S is L_s. As for bounding the AMF values, for each $y \in [0, 1]$, if we let $\frac{i}{m}$ be the element of $\{0,\, 1/m,\, 2/m,\, \ldots,\, 1\}$ closest to y, then $|y - \frac{i}{m}| \le \frac{1}{2m}$. Define $x \in \Delta_{m+1}$ to be a point-mass distribution on i. Then,

$$\|\mathbf{c}(x, y)\|_1 = \left|y - \frac{i}{m}\right| \le \frac{1}{2m},$$

Meanwhile,

$$r_t(x, y) = S\left(\frac{i}{m}, y\right) - S\left(q_t, y\right) \le S\left(\frac{i}{m}, y\right) - S\left(y, y\right) \le L_s \left|\frac{i}{m} - y\right| \le \frac{L_s}{2m}.$$

Hence,

$$w_A^t = \sup_{y\in[0,1]} \min_{x\in\Delta_{m+1}} \left\{\max_{j\in[d]} \tilde{\ell}_j^t(x, y)\right\} = \sup_{y\in[0,1]} \min_{x\in\Delta_{m+1}} \left\{\max\left(\|\mathbf{c}(x, y)\|_1,\, r_t(x, t)\right)\right\} \le \frac{\max(1, L_s)}{2m}.$$

Using the upper bounds $C \le \max(1, L_s)$ and $w_A^t \le \frac{\max(1,L_s)}{2m}$ in Theorem 14, we find that if the algorithm of Lee et al. (2022) with learning rate $\eta = \sqrt{\frac{\ln d}{4CT^2}}$ is applied to the sequence of environments $(X^t, Y^t, \tilde{\ell}^t)$ it will satisfy the bound

$$\max_{j\in[d]} \left(\frac{1}{T}\sum_{t=1}^{T} \tilde{\ell}^t{}_j(x^t, y^t)\right) \le \frac{1}{T}\sum_{t=1}^{T} w_A^t \;+\; \frac{1}{T}\cdot 4C\sqrt{T\ln d} \le \max(1, L_s)\cdot\left(\frac{1}{2m} \;+\; 4\sqrt{\frac{\ln d}{T}}\right).$$

Earlier we derived that the left side is the maximum of the forecaster's ℓ_1 calibration error and average regret. Recalling that $1/m = \varepsilon$ and that $d = 2^{m+1} + 1$, we find that both the ℓ_1 calibration error and the average regret are bounded above by $\max(1, L_s)\cdot O(\varepsilon + 1/\sqrt{\varepsilon T}))$.

Compared to this bound, our Algorithm 1 achieves the same upper bound on ℓ_1 calibration error but an improved bound of $O(L_s\varepsilon^2 + 1/\sqrt{\varepsilon T})$ on average regret. It is tempting to try to modify the reduction from recalibration to online multiobjective optimization, to see if it can achieve the same bound. For example, above when we derived the inequality $r_t(x, y) \le \frac{L_s}{2m}$, a more refined analysis using the property that the scoring rule S is strictly

proper would yield the bound $r_t(x,y) \le O\left(\frac{L_s}{m^2}\right)$. This means one could modify the definition of the loss vectors $\tilde{\ell}^t(x,y)$ by rescaling their final coordinate to equal $m \cdot r_t(x,y)$ rather than $r_t(x,y)$, without invalidating the upper bound on the AMF values w_A^t. Then an upper bound of the form $\max_j \left(\frac{1}{T}\sum_t \tilde{\ell}_j^t(x^t,y^t)\right) \le O(\varepsilon)$ would simultaneously imply ℓ_1 calibration error $O(\varepsilon)$ and average regret $O(\varepsilon^2)$, because one gains a factor of $1/m = \varepsilon$ when rescaling the final coordinate of $\frac{1}{T}\sum_t \tilde{\ell}_j^t(x^t,y^t)$ to convert it back into average regret. However, defining the final coordinate of $\tilde{\ell}^t(x,y)$ to equal $m \cdot r_t(x,y)$ would mean that the infinity-norm of the loss vectors is bounded above by $C = m \cdot \max(1, L_s)$, it is no longer bounded above merely by $\max(1, L_s)$. Hence, the rescaling inflates the regret term $4C\sqrt{T \ln d}$ in Theorem 14 by a factor of $m = 1/\varepsilon$, more than offsetting any potential gains resulting from the rescaling.

B.3 Efficient Implementation of the Reduction

Because the reduction described in Section B.2 involves loss vectors in dimension $d = 2^{m+1} + 1$, a straightforward implementation of the reduction runs the risk of requiring running time $O(2^m)$ per iteration. Fortunately, there is an implementation requiring only $\text{poly}(m)$ running time per iteration. The key to avoiding the exponential dependence on m is, first of all, to store the vectors $\ell^t(x^t,y^t)$, which are only $(m+2)$-dimensional, rather than the exponentially higher-dimensional loss vectors $\tilde{\ell}(x^t,y^t)$. However, the algorithm still needs to compute

$$x^t \in \arg\min_{x \in X^t} \max_{y \in Y^t} \langle \chi^t, \tilde{\ell}^t(x,y) \rangle$$

where χ^t is a d-dimensional vector with coordinates

$$\chi_j^t = \frac{\exp(\eta \sum_{s=1}^{t-1} \tilde{\ell}_j^s(x^s,y^s))}{\sum_{i \in [d]} \exp(\eta \sum_{s=1}^{t-1} \tilde{\ell}_i^s(x^s,y^s))}.$$

Expanding out the inner product $\langle \chi^t, \tilde{\ell}^t(x,y) \rangle$ in the definition of x^t, we find that

$$x^t \in \arg\min_{x \in X^t} \max_{y \in Y^t} \frac{\sum_{j \in [d]} \exp(\eta \sum_{s=1}^{t-1} \tilde{\ell}_j^s(x^s,y^s)) \cdot \tilde{\ell}_j^t(x^t,y^t)}{\sum_{i \in [d]} \exp(\eta \sum_{s=1}^{t-1} \tilde{\ell}_i^s(x^s,y^s))}. \tag{36}$$

To compute the sums in the numerator and denominator, recall that for each $j \in [d-1]$ there is a corresponding sign vector $\sigma \in \{\pm 1\}^{m+1}$ such that $\tilde{\ell}_j^s(x^s,y^s) = \sum_{k=1}^{m+1} \sigma_k \ell_k^s(x^s,y^s)$. Hence, the sum in the denominator of Equation (36) simplifies as

$$\begin{aligned}
\sum_{i \in [d]} \exp(\eta \sum_{s=1}^{t-1} \tilde{\ell}_i^s(x^s,y^s)) &= \exp(\eta \sum_{s=1}^{t-1} \tilde{\ell}_d^s(x^s,y^s)) + \sum_{\sigma \in \{\pm 1\}^{m+1}} \exp(\eta \sum_{s=1}^{t-1} \sum_{k=1}^{m+1} \sigma_k \ell_k^s(x^s,y^s)) \\
&= \exp(\eta \sum_{s=1}^{t-1} \tilde{\ell}_d^s(x^s,y^s)) + \sum_{\sigma \in \{\pm 1\}^{m+1}} \prod_{k=1}^{m+1} \exp(\eta \sigma_k \sum_{s=1}^{t-1} \ell_k^s(x^s,y^s)) \\
&= \exp(\eta \sum_{s=1}^{t-1} \tilde{\ell}_d^s(x^s,y^s)) + \prod_{k=1}^{m+1} \sum_{\sigma_k \in \pm 1} \exp(\eta \sigma_k \sum_{s=1}^{t-1} \ell_k^s(x^s,y^s))
\end{aligned} \tag{37}$$

and the formula on the last line can be computed in $O(mt)$ arithmetic operations. In fact, a further running time improvement can be achieved by using dynamic programming to amortize over time steps. At the end of each time step, if we store the quantities $\exp(\eta \sum_{s=1}^{t-1} \tilde{\ell}_d^s(x^s,y^s))$ and $\exp(\eta \sigma_k \sum_{s=1}^{t-1} \ell_k^s(x^s,y^s))$ for each $k \in [m+1]$ and $\sigma_k \in \pm 1$, then updating these values to incorporate the loss vector from time $s = t$ requires constant time (a single multiplicative update) for each of the $2m+3$ stored values. Evaluating the formula on the last line of Equation (37) then requires applying only $O(m)$ arithmetic operations to the stored values.

A similar simplification pertains to the numerator in Equation (36). To save space, we will ignore the $j = d$ term of the sum, which is a special case that can be computed separately from the terms corresponding to $j \in [d-1]$.

As before, each $j \in [d-1]$ corresponds to a sign vector $\sigma \in \{\pm 1\}^{m+1}$.

$$\begin{aligned}
&\sum_{\sigma \in \{\pm 1\}^{m+1}} \exp\left(\eta \sum_{s=1}^{t-1} \sum_{k=1}^{m+1} \sigma_k \ell_k^s(x^s, y^s)\right) \left(\sum_{k'=1}^{m+1} \sigma_{k'} \ell_{k'}^t(x^t, y^t)\right) \\
&= \sum_{k'=1}^{m+1} \sum_{\sigma \in \{\pm 1\}^{m+1}} \prod_{k=1}^{m+1} \exp\left(\eta \sum_{s=1}^{t-1} \sigma_k \ell_k^s(x^s, y^s)\right) \sigma_{k'} \ell_{k'}^t(x^t, y^t) \\
&= \sum_{k'=1}^{m+1} \left(\sum_{\sigma_{k'} \in \{\pm 1\}} \exp\left(\eta \sigma_{k'} \sum_{s=1}^{t-1} \ell_{k'}^s(x^s, y^s)\right) \sigma_{k'} \ell_{k'}^t(x^t, y^t) \right) \cdot \prod_{k \neq k'} \left(\sum_{\sigma_k \in \{\pm 1\}} \exp\left(\eta \sigma_k \sum_{s=1}^{t-1} \ell_k^s(x^s, y^s)\right) \right). \quad (38)
\end{aligned}$$

As before, using dynamic programming the expression on the last line can be computed using only $O(m)$ arithmetic operations per time step.