High-Dimensional Calibration from Swap Regret

Anonymous Author(s)
Affiliation
Address
email

Abstract

We study the online calibration of multi-dimensional forecasts over an arbitrary convex set $\mathcal{P} \subset \mathbb{R}^d$ relative to an arbitrary norm $\|\cdot\|$. We connect this with the problem of external regret minimization for online linear optimization, showing that if it is possible to guarantee $O(\sqrt{\rho T})$ worst-case regret after T rounds when actions are drawn from $\mathcal{P}$ and losses are drawn from the dual $\|\cdot\|_*$ unit norm ball, then it is also possible to obtain ϵ-calibrated forecasts after $T = \exp(O(\rho/\epsilon^2))$ rounds. When $\mathcal{P}$ is the d-dimensional simplex and $\|\cdot\|$ is the ℓ_1-norm, the existence of $O(\sqrt{T \log d})$ algorithms for learning with experts implies that it is possible to obtain ϵ-calibrated forecasts after $T = \exp(O(\log d/\epsilon^2)) = d^{O(1/\epsilon^2)}$ rounds, recovering a recent result of [Pen25].

Interestingly, our algorithm obtains this guarantee without requiring access to any online linear optimization subroutine or knowledge of the optimal rate ρ – in fact, our algorithm is identical for every setting of $\mathcal{P}$ and $\|\cdot\|$. Instead, we show that the optimal regularizer for the above OLO problem can be used to upper bound the above calibration error by a swap regret, which we then minimize by running the recent TreeSwap algorithm ([DDFG24, PR24]) with Follow-The-Leader as a subroutine. The resulting algorithm is highly efficient and plays a distribution over simple averages of past observations in each round.

Finally, we prove that any online calibration algorithm that guarantees ϵT ℓ_1-calibration error over the d-dimensional simplex requires $T \geq \exp(\mathrm{poly}(1/\epsilon))$ (assuming $d \geq \mathrm{poly}(1/\epsilon)$). This strengthens the corresponding $d^{\Omega(\log 1/\epsilon)}$ lower bound of [Pen25], and shows that an exponential dependence on $1/\epsilon$ is necessary.

1 Introduction

Consider the problem faced by a forecaster who must report probabilistic predictions for a sequence of events (e.g. whether it will rain or not tomorrow). One of the most common methods to evaluate the quality of such a forecaster is to verify whether they are *calibrated*: for example, does it indeed rain with probability 40% on days where the forecaster makes this prediction? In addition to calibration being a natural property to expect from predictions, several applications across machine learning, fairness, and game theory require the ability to produce online calibrated predictions [ZME20, GPSW17, HJKRR18, FV97].

When events have binary outcomes, calibration can be quantified by the notion of *expected calibration error*, which measures the expected distance between a prediction made by a forecaster and the actual empirical probability of the outcome on the days where they made that prediction. In a seminal result by Foster and Vohra [FV98], it was proved that it is possible for an online forecaster to efficiently guarantee a sublinear calibration error of $O(T^{2/3})$ against any adversarial sequence of T binary events. Equivalently, this can be interpreted as requiring at most $O(\epsilon^{-3})$ rounds of forecasting to guarantee an ϵ per-round calibration error on average.

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

However, many applications require forecasting sequences of *multi-dimensional* outcomes. The previous definition of calibration error easily extends to the multi-dimensional setting where predictions and outcomes belong to a d-dimensional convex set $\mathcal{P} \subset \mathbb{R}^d$. Specifically, if a forecaster makes a sequence of predictions $p_1, p_2, \ldots, p_T \in \mathcal{P}$ for the outcomes $y_1, y_2, \ldots, y_T \in \mathcal{P}$, their $\|\cdot\|$-calibration error (for any norm $\|\cdot\|$ over $\mathbb{R}^d$) is given by

$$\mathsf{Cal}_T^{\|\cdot\|} = \sum_{t=1}^{T} \|p_t - \nu_{p_t}\|$$

where ν_{p_t} is the average of the outcomes y_t on rounds where the learner predicted p_t.

The algorithm of Foster and Vohra extends to the multidimensional calibration setting, but at the cost of producing bounds that decay exponentially in the dimension d. In particular, their algorithm only guarantees that the forecaster achieves an average calibration error of ϵ after $(1/\epsilon)^{\Omega(d)}$ rounds. Until recently, no known algorithm achieved a sub-exponential dependence on d in any non-trivial instance of multi-dimensional calibration.

In 2025, [Pen25] presented a new algorithm for high-dimensional calibration, demonstrating that it is possible to obtain ℓ_1-calibration rates of ϵT in $d^{O(1/\epsilon^2)}$ rounds for predictions over the d-dimensional simplex (i.e., multi-class calibration). In particular, this is the first known algorithm achieving polynomial calibration rates in d for fixed constant ϵ. [Pen25] complements this with a lower bound, showing that in the worst case $d^{\Omega(\log 1/\epsilon)}$ rounds are necessary to obtain this rate (implying that a fully polynomial bound $\mathrm{poly}(d, 1/\epsilon)$ is impossible).

1.1 Our results

Although the algorithm of [Pen25] is simple to describe, its analysis is fairly nuanced and tailored to ℓ_1-calibration over the simplex (e.g., by analyzing the KL divergence between predictions and distributions of historical outcomes). We present a very similar algorithm (`TreeCal`) for multi-dimensional calibration over an arbitrary convex set $\mathcal{P} \subset \mathbb{R}^d$, but with a simple, unified analysis that provides simultaneous guarantees for calibration with respect to any norm $\|\cdot\|$. In particular, we prove the following theorem.

Theorem 1.1 (Informal restatement of Corollary C.5)**.** *Fix a convex set $\mathcal{P}$ and a norm $\|\cdot\|$. Assume there exists a function $R : \mathcal{P} \to \mathbb{R}$ that is 1-strongly-convex with respect to $\|\cdot\|$ and has range $(\max_{x\in\mathcal{P}} R(x) - \min_{p\in\mathcal{P}} R(x))$ at most ρ. Then* `TreeCal` *guarantees that the calibration error of its predictions is bounded by $\mathsf{Cal}_T^{\|\cdot\|} \le \epsilon T$ for $T \ge (\mathsf{diam}_{\|\cdot\|}(\mathcal{P})/\epsilon)^{O(\rho/\epsilon^2)}$.*

Interestingly, the function $R(p)$ and parameter ρ appearing in the statement of Theorem 1.1 have an independent learning-theoretic interpretation: if we consider the *online linear optimization* problem where a learner plays actions in $\mathcal{P}$ and the adversary plays linear losses that are unit bounded in the dual norm $\|\cdot\|_*$, then it is possible for the learner to guarantee a regret bound of at most $O(\sqrt{\rho T})$ by playing Follow-The-Regularized-Leader (FTRL) with $R(p)$ as a regularizer. In fact, since universality results for mirror descent guarantee that some instantiation of FTRL achieves near-optimal rates for online linear optimization (as long as the action and loss sets are centrally convex) [SST11, GSJ24], this allows us to relate the performance of Theorem 3.1 directly to what rates are possible in online linear optimization.

Corollary 1.2 (Informal restatement of Corollary C.6)**.** *Let $\mathcal{P} \subseteq \mathbb{R}^d$ be a centrally symmetric convex set, and let $\mathcal{L} = \{y \in \mathbb{R}^d \mid \|y\|_* \le 1\}$ for some norm $\|\cdot\|$. Then if there exists an algorithm for online linear optimization with action set $\mathcal{P}$ and loss set $\mathcal{L}$ that incurs regret at most $O(\sqrt{\rho T})$,* `TreeCal` *guarantees that the calibration error of its predictions is bounded by $\mathsf{Cal}_T^{\|\cdot\|} \le \epsilon T$ for $T \ge (\mathsf{diam}_{\|\cdot\|}(\mathcal{P})/\epsilon)^{O(\rho/\epsilon^2)}$.*

Theorem 1.1 and its corollary allow us to immediately recover several existing and novel bounds on calibration error in a variety of settings:

- When $\mathcal{P}$ is the d-simplex Δ_d and $\|\cdot\|$ is the ℓ_1-norm, the existence of the negative entropy regularizer $R(x) = \sum_{i=1}^{d} x_i \log x_i$ (which is 1-strongly convex w.r.t. the ℓ_1 norm with range

$\rho = \log d$) implies that the ℓ_1 calibration error of `TreeCal` is at most $(1/\epsilon)^{O(\log d/\epsilon^2)} = d^{\tilde{O}(1/\epsilon^2)}$. This recovers the result of [Pen25].

- When $\mathcal{P}$ is the ℓ_2 ball and $\|\cdot\|$ is the ℓ_2 norm, the Euclidean regularizer ($R(x) = \|x\|^2$) implies a calibration bound of $(1/\epsilon)^{O(1/\epsilon^2)}$ (notably, this bound is independent of d).

It should be emphasized here that running `TreeCal` does not require any online linear optimization subroutine, nor any knowledge of these regularizers $R(x)$ or optimal rates ρ. `TreeCal` has no functional dependence on any specific $\|\cdot\|$. It achieves $\|\cdot\|$-calibration at the above rate (Theorem 1.1) for all $\|\cdot\|$ simultaneously. The `TreeCal` algorithm is nearly identical[1] to the algorithm of [Pen25] – both algorithms initialize a tree of sub-forecasters and at each round play a uniform combination of some subset of them (see Figure 1).

The novelty in our analysis stems from the observation that `TreeCal` is simply a specific instantiation of the `TreeSwap` swap regret minimization algorithm [DDFG24, PR24] and can be analyzed directly in this way. In particular, our analysis consists of the following steps:

1. First, minimizing calibration error can be reduced to minimizing swap regret, generalizing an idea of [LSS25, FKO$^+$25]. That is, it is possible to assign the learner loss functions $\ell_t : \mathcal{P} \to \mathbb{R}$ at each round such that their calibration error is upper bounded by the gap between the total loss they received, and the minimal loss they could have received after applying an arbitrary "swap function" $\pi : \mathcal{P} \to \mathcal{P}$ to their predictions.

   In fact, any strongly convex function R (w.r.t. the norm $\|\cdot\|$) gives rise to one such reduction, by setting the loss function $\ell_t(p)$ to equal the Bregman divergence $D_R(y_t|p)$.
2. Second, the `TreeSwap` algorithm of [DDFG24, PR24] provides a general recipe for converting external regret minimization algorithms into swap regret minimization algorithms. We obtain `TreeCal` by plugging in the Follow-The-Leader algorithm (the learning algorithm which simply always best responds to the current history) into `TreeSwap`.
3. Instead of analyzing the swap regret bound of `TreeSwap` with Follow-The-Leader (which may not have a good enough external regret bound, as discussed in Section 3.3), we instead analyze the swap regret of `TreeSwap` with *Be-The-Leader* (the fictitious algorithm that best responds to the current history, including the current round). Though it is not possible to actually implement Be-The-Leader due to its clairvoyance, we use it as a tool for analysis. We then relate the calibration error of `TreeSwap` with *Be-The-Leader* to that of `TreeSwap` with *Follow-The-Leader* using the fact that Be-The-Leader and Follow-The-Leader make similar predictions.

In the above step 1, we will choose R to be $\|\cdot\|$-norm 1-strongly convex, which guarantees that $D_R(y|p) \geq \|y - p\|^2$. Going through the analysis, this actually leads to the stronger guarantee that `TreeCal` minimizes *squared-norm* calibration error.

Theorem 1.3 (Informal restatement of Theorem 3.1)**.** *Fix a convex set $\mathcal{P}$ and a norm $\|\cdot\|$. Assume there exists a function $R : \mathcal{P} \to \mathbb{R}$ that is 1-strongly-convex with respect to $\|\cdot\|$ and has range $(\max_{x\in\mathcal{P}} R(x) - \min_{p\in\mathcal{P}} R(x))$ at most ρ. Then `TreeCal` guarantees that the calibration error of its predictions is bounded by $\mathsf{Cal}_T^{\|\cdot\|^2} \leq \epsilon T$ for $T \geq (\mathsf{diam}_{\|\cdot\|}(\mathcal{P})/\sqrt{\epsilon})^{O(\rho/\epsilon)}$.*

Note here we have only singly-exponential dependence on $1/\epsilon$. We arrive at Theorem 1.1 as a corollary of this result by simply applying Cauchy-Schwarz. Finally, we strengthen the lower bound of [Pen25] by showing an exponential dependence on $1/\epsilon$ is necessary.

Theorem 1.4 (Informal restatement of Theorem 4.3)**.** *There is a sufficiently small constant $c > 0$ so that the following holds. Fix any $\epsilon > 0, d \in \mathbb{N}$. Then for any $T \leq \exp(c \cdot \min\{d^{1/14}, \epsilon^{-1/6}\})$, there is an oblivious adversary producing a sequence of outcomes so that any learning algorithm must incur ℓ_1-calibration error $\mathsf{Cal}_T^{\|\cdot\|_1} \geq \epsilon \cdot T$.*

Unlike the lower bound of [Pen25], this lower bound requires no specialized construction. Instead, it follows from the original observation of [FV98] that any algorithm for online calibration can be

[1] One minor difference is that the algorithm of [Pen25] regularizes each sub-forecaster by slightly mixing their prediction with the uniform distribution, which `TreeCal` does not require.

used to construct an algorithm for swap regret minimization by simply best responding to a sequence of calibrated predictions of the adversary's losses. The existing lower bound for swap regret in [DFG+24] then immediately precludes the existence of sufficiently strong calibration bounds (e.g., of the form $d^{O(\log 1/\epsilon)}$, which was still allowed by the work of [Pen25]). We discuss additional related work in the appendix.

2 Setup

For a positive integer n, we let $[0 : n-1]$ denote the sequence $0, 1, \ldots, n-1$, and $[n]$ denote the sequence $1, 2, \ldots, n$. We say a convex set $\mathcal{S} \subseteq \mathbb{R}^d$ is *centrally symmetric* if $s \in \mathcal{S} \Leftrightarrow -s \in \mathcal{S}$ for all $s \in \mathbb{R}^d$. A norm $\|\cdot\|$ is a function corresponding to a convex, bounded, centrally-symmetric set $\mathcal{S}$ of the form $\|s\| = \inf \{c \in \mathbb{R}_{\geq 0} | s \in c\mathcal{S}\}$. The corresponding *dual norm* is defined $\|v\|_* = \sup \{\langle s, v\rangle \mid \|s\| \leq 1\}$.

2.1 Calibration

We consider the following setting of *multi-dimensional calibration*. Positive integers $d \in \mathbb{N}$ representing the number of dimensions and $T \in \mathbb{N}$ representing the number of rounds are given. We let $\mathcal{P} \subset \mathbb{R}^d$ denote a bounded convex subset of $\mathbb{R}^d$. An *adversary* and a *learning algorithm* interact for a total of T timesteps; at each time step $t \in [T]$:

- The learning algorithm chooses a distribution[2] $\mathbf{x}_t \in \Delta(\mathcal{P})$ with finite support.
- The adversary observes $\mathbf{x}_t$ and chooses an *outcome* $y_t \in \mathcal{P}$.

In order for the learner to be calibrated, we would like the average outcome conditional on the learner making a specific prediction p to be "close" to p. We formalize this as follows. For a point $p \in \mathcal{P}$, we define ν_p to be the average outcome conditioned on the learner predicting p, that is:

$$\nu_p := \frac{\sum_{t=1}^{T} \mathbf{x}_t(p) \cdot y_t}{\sum_{t=1}^{T} \mathbf{x}_t(p)}. \tag{1}$$

Fix a *distance measure* $D : \mathcal{P} \times \mathcal{P} \to \mathbb{R}_{\geq 0}$, namely an arbitrary non-negative valued function on $\mathcal{P} \times \mathcal{P}$. Given a distance measure D, we define the *D-calibration error* as follows:

$$\mathsf{Cal}_T^D(\mathbf{x}_{1:T}, y_{1:T}) := \sum_{p \in \mathcal{P}} \left(\sum_{t=1}^{T} \mathbf{x}_t(p) \right) \cdot D(\nu_p, p).$$

In the event that $D(p, q) = \|p - q\|$, we will write $\mathsf{Cal}_T^{\|\cdot\|}(\mathbf{x}_{1:T}, y_{1:T}) = \mathsf{Cal}_T^D(\mathbf{x}_{1:T}, y_{1:T})$, and we define $\mathsf{Cal}_T^{\|\cdot\|^2}(\mathbf{x}_{1:T}, y_{1:T})$ analogously.

2.2 Regret minimization

For a sequence of actions $p_1, \cdots, p_T \in \mathcal{P}$ and loss functions $\ell_1, \cdots, \ell_T : \mathcal{P} \to \mathbb{R}$, we define

$$\mathsf{ExtReg}_T(p_{1:T}, \ell_{1:T}) := \sup_{p^* \in \mathcal{P}} \sum_{t=1}^{T} \sum_{p \in \mathcal{P}} \ell_t(p_t) - \ell_t(p^*)$$

For a sequence of distributions $\mathbf{x}_1, \cdots, \mathbf{x}_T \in \Delta(\mathcal{P})$ and loss functions $\ell_1, \cdots, \ell_T : \mathcal{P} \to \mathbb{R}$, we define

$$\mathsf{FullSwapReg}_T(\mathbf{x}_{1:T}, \ell_{1:T}) := \sup_{\pi : \mathcal{P} \to \mathcal{P}} \sum_{t=1}^{T} \sum_{p \in \mathcal{P}} \mathbf{x}_t(p) \cdot (\ell_t(p) - \ell_t(\pi(p))). \tag{2}$$

[2] Some authors refer to this setting as "pseudo-calibration" or "distributional calibration", and reserve the term "calibration" for the setting where the learner is required to randomly select a pure forecast $p_t \in \mathcal{P}$ each round instead of a distribution. In Appendix E we describe how to extend our results to this pure-strategy setting of calibration.

Here, we adopt the convention of [FKO+25], referring to the latter quantity as *Full* Swap Regret to emphasize that we consider *all* swap transformations $\pi : \mathcal{P} \to \mathcal{P}$ (instead of e.g. just linear transformations π).

Throughout, we consider the performance of *regret minimizing* algorithms. These algorithms sequentially map loss functions $\ell_1, \cdots, \ell_T$ to actions $p_1, \cdots, p_T$ or action distributions $\mathbf{x}_1, \cdots, \mathbf{x}_T$ with the goal of minimizing the above quantities. We consider the performance of these algorithms on adversarially selected loss functions from a set $\mathcal{L}$. Abusing notation slightly, for an external regret minimizing algorithm $\mathtt{Alg} : \mathcal{L}^T \to \mathcal{P}^T$, we define

$$\mathsf{ExtReg}_T(\mathtt{Alg}) := \sup_{\ell_{1:T} \in \mathcal{L}^T} \mathsf{ExtReg}_T\left(\mathtt{Alg}(\ell_{1:T}), \ell_{1:T}\right) \tag{3}$$

and for a full swap regret minimizing algorithm $\mathtt{Alg} : \mathcal{L}^T \to \Delta(\mathcal{P})^T$, we define

$$\mathsf{FullSwapReg}_T(\mathtt{Alg}) := \sup_{\ell_{1:T} \in \mathcal{L}^T} \mathsf{FullSwapReg}_T\left(\mathtt{Alg}(\ell_{1:T}), \ell_{1:T}\right).$$

We will denote the tth action played by $\mathtt{Alg}$ on a sequence of losses $\ell_{1:T}$ by $\mathtt{Alg}_t(\ell_{1:T})$. One important subclass of external regret minimization problems is the setting of *online linear optimization (OLO)*, where all loss functions in ℓ are linear. Here we slightly abuse notation and identify $\mathcal{L}$ with a subset of $\mathbb{R}^d$ (with the understanding that an element $\ell \in \mathcal{L}$ refers to the linear loss function $\ell(p) = \langle p, \ell \rangle$). Although we will never actually employ any OLO algorithms themselves, the calibration bounds we obtain will be closely related to optimal regret bounds for instances of OLO (we discuss this further in Section 2.4).

2.3 From swap regret to calibration

As noted in [LSS25, FKO+25], calibration with a distance measure D that corresponds to a *Bregman divergence* can be written as a full swap regret with loss functions given by the associated *proper scoring rule*. Given a convex function $R : \mathcal{P} \to \mathbb{R}$, the *Bregman divergence* associated to R, $D_R : \mathcal{P} \times \mathcal{P} \to \mathbb{R}_{\geq 0}$, is defined as[3]

$$D_R(y|p) := R(y) - R(p) - \langle \nabla R(p), y - p \rangle$$

Geometrically, this divergence is defined by taking the hyperplane tangent to R at p and computing the difference in height between R and the hyperplane at y (see Figure 2).

When viewed as a loss function in p, the Bregman divergence $D_R(y|p)$ also has the property that it is a *proper scoring rule*. This refers to the fact that if y is drawn from some distribution $\mathbf{y} \in \Delta(\mathcal{P})$, the optimal response p (to minimize the expected loss $D_R(y|p)$) is simply the expectation $\bar{y} = \mathbb{E}_{y\sim\mathbf{y}}[y]$. In particular, we have the following lemma.

Lemma 2.1. *For any $\mathbf{y} \in \Delta(\mathcal{P})$ and convex function $R : \mathcal{P} \to \mathbb{R}$, let $\bar{y} = \mathbb{E}_{y\sim\mathbf{y}}[y]$. and $\overline{R(y)} = \mathbb{E}_{y\sim\mathbf{y}}[R(y)]$. For all $p \in \mathcal{P}$, $\mathbb{E}_{y\sim\mathbf{y}}[D_R(y|p)] = D_R(\bar{y}|p) + \overline{R(y)} - R(\bar{y})$. In particular, $\ell(p) = \mathbb{E}_{y\sim\mathbf{y}}[D_R(y|p)]$ is minimized at $p = \bar{y}$ at a value of $\overline{R(y)} - R(\bar{y})$ (Figure 3).*

This implies the following connection between full swap regret and calibration.

Lemma 2.2. *Fix any convex function $R : \mathcal{P} \to \mathbb{R}$. For any sequence of distributions $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T \in \Delta(\mathcal{P})$ and outcomes $y_1, y_2, \ldots, y_T \in \mathcal{P}$, define the sequence of loss functions $\ell_1, \ell_2, \ldots, \ell_T$ via $\ell_t(p) = D_R(y_t|p)$. Then,*

$$\mathsf{FullSwapReg}_T(\mathbf{x}_{1:T}, \ell_{1:T}) = \mathsf{Cal}_T^{D_R}(\mathbf{x}_{1:T}, y_{1:T}).$$

2.4 Rates and regularization

In order to reduce our general calibration problem to a swap regret minimization problem (via Lemma 2.2), we will need to construct a convex function R whose Bregman divergence upper bounds our distance measure. It turns out that the optimal choice of such a function is closely related to

[3] In the event that R is not differentiable, we can replace the $\nabla R(p)$ term with any element of the sub-gradient at p. When $\mathcal{P}$ is not open and p is on the boundary, the $\nabla R(p)$ term represents the inward directional gradient.

the design of optimal regularizers for online linear optimization. In this section, we describe this functional optimization problem and detail this connection.

We say that a convex function $R : \mathcal{P} \to \mathbb{R}$ is α-*strongly convex* with respect to a given norm $\|\cdot\|$ if for any points $y, p \in \mathcal{P}$ it is the case that $R(y) \geq R(p) + \langle \nabla R(p), y - p \rangle + \alpha \|y - p\|^2$. Equivalently, the Bregman divergence must satisfy $D_R(y|p) \geq \alpha \|y - p\|^2$. Thus, $\|\cdot\|^2$-calibration error is bounded by D_R-calibration error if R is $\|\cdot\|$-norm 1-strongly convex.

Our later analysis will need not only R to be strongly convex with respect to our norm, but for the Bregman divergence to have a small maximal value. Motivated by this, we will say that a convex function $R : \mathcal{P} \to \mathbb{R}$ has *rate* ρ with respect to a given norm $\|\cdot\|$ if: **(1)** R is 1-strongly convex with respect to $\|\cdot\|$, and **(2)** the range of the Bregman divergence is at most ρ, i.e., $\max_{y,p \in \mathcal{P}} D_R(y|p) \leq \rho$. We define $\mathsf{Rate}(\mathcal{P}, \|\cdot\|)$ to be the infimum of the rates of all 1-strongly convex functions $R : \mathcal{P} \to \mathbb{R}$.

As mentioned earlier, we call this quantity a "rate" due to its connection with the optimal regret rates for online linear optimization. For a learning algorithm $\texttt{Alg} : \mathcal{L}^T \to \mathcal{P}^T$, we defined (in (3)) $\mathsf{ExtReg}_T(\texttt{Alg})$ to be the worst-case regret against any sequence $\ell_{1:T}$ of T losses. It is known that for any fixed action set and loss set, the optimal worst-case regret bound is of the form $\sqrt{\mathsf{Rate}_{\mathsf{OLO}}(\mathcal{P}, \mathcal{L}) \cdot T} + o(\sqrt{T})$, for some constant $\mathsf{Rate}_{\mathsf{OLO}}(\mathcal{P}, \mathcal{L})$. Formally, we define $\mathsf{Rate}_{\mathsf{OLO}}(\mathcal{P}, \mathcal{L}) = \limsup_{T \to \infty} \inf_{\texttt{Alg}} \frac{1}{T} \cdot \mathsf{ExtReg}_T(\texttt{Alg})^2$.

One important class of learning algorithms for online linear optimization is the class of Follow-The-Regularized-Leader (FTRL) algorithms. Each algorithm in this class is specified by a convex "regularizer" function $R : \mathcal{P} \to \mathbb{R}$, and at round t selects the action $p_t = \operatorname{argmin}_{p \in \mathcal{P}} \sum_{s=1}^{t-1} \langle p, \ell_t \rangle + R(p)$. The work of [SST11] and [GSJ24] shows that there always exists some instantiation of FTRL which achieves (up to a universal constant factor) the optimal regret rate of $\sqrt{\mathsf{Rate}_{\mathsf{OLO}}(\mathcal{P}, \mathcal{L}) \cdot T} + o(\sqrt{T})$ defined above. Moreover, the optimal regularizer for this instance can be constructed by solving a similar functional optimization problem over strongly convex regularizers R, as described in the following theorem.

Theorem 2.3. *Let $\mathcal{P}$ and $\mathcal{L}$ be centrally symmetric convex sets. Then, if the function $R : \mathcal{P} \to \mathbb{R}$ is 1-strongly-convex with respect to the norm $\|\cdot\|_{\mathcal{L}^*}$ and has range ρ (i.e., $\max_{p \in \mathcal{P}} R(p) - \min_{p \in \mathcal{P}} R(p) = \rho$), then $\mathsf{Rate}_{\mathsf{OLO}}(\mathcal{P}, \mathcal{L}) \leq \rho$. Conversely, there exists a function $R : \mathcal{P} \to \mathbb{R}$ that is 1-strongly-convex with respect to $\|\cdot\|_{\mathcal{L}^*}$ and has range $O(\mathsf{Rate}_{\mathsf{OLO}}(\mathcal{P}, \mathcal{L}))$.*

Proof. The first result (that $\mathsf{Rate}_{\mathsf{OLO}}(\mathcal{P}, \mathcal{L}) \leq \rho$) follows from the standard analysis of FTRL – see e.g. Theorem 5.2 in [H+16]. The converse result follows from Theorem 2 of [GSJ24]. □

Theorem 2.3 allows us to relate the quantity $\mathsf{Rate}(\mathcal{P}, \|\cdot\|)$ to the quantity $\mathsf{Rate}_{\mathsf{OLO}}(\mathcal{P}, \mathcal{L})$ (where $\mathcal{L}$ is chosen to be the unit dual norm ball). Note that there is a slight difference in the two functional optimization problems defined above – the one for $\mathsf{Rate}(\mathcal{P}, \|\cdot\|)$ asks us to bound the range of the Bregman divergence of R, while the one for $\mathsf{Rate}_{\mathsf{OLO}}(\mathcal{P}, \mathcal{L})$ asks us to bound the range of R itself. While these two quantities do not directly bound each other (the negative entropy function $R(p) = \sum p_i \log p_i$ has bounded range over the simplex but unbounded Bregman divergence), we can nonetheless show that optimal solutions to one problem can be used to construct optimal solutions to the other problem of similar quality.

Lemma 2.4. *If the action set $\mathcal{P}$ is centrally symmetric and $\mathcal{L} = \{y \in \mathbb{R}^d \mid \|y\|_* \leq 1\}$ (i.e., the unit ball in the dual norm to $\|\cdot\|$), then $\mathsf{Rate}_{\mathsf{OLO}}(\mathcal{P}, \mathcal{L}) = \Theta(\mathsf{Rate}(\mathcal{P}, \|\cdot\|))$.*

3 Main result

We now describe our main algorithm for calibration, `TreeCal` (Algorithm 1). As we will see, it is equivalent to the `TreeSwap` algorithm for Full Swap Regret minimization ([DDFG24, PR24]; Algorithm 2), where the loss functions are given by appropriate Bregman divergences as determined by Lemma 2.2. Moreover, `TreeCal` is effectively the same as the main algorithm of [Pen25]. However, the perspective that `TreeCal` can be viewed as a particular instance of `TreeSwap` (Lemma 3.2) is novel to this work, and it enables us to tackle a much more general set of calibration problems (Theorem 3.1). We first describe the `TreeCal` and `TreeSwap` algorithms, then state Theorem 3.1 which establishes our main upper bound for `TreeCal`, and finally discuss the proof of Theorem 3.1, which uses the `TreeSwap` algorithm as a tool in the analysis.

3.1 Algorithm description

Given some number of rounds $T \in \mathbb{N}$, `TreeCal` and `TreeSwap` sequentially produce distributions $\mathbf{x}_1, \cdots, \mathbf{x}_T \in \Delta(\mathcal{P})$. `TreeCal` receives from the adversary an outcome sequence $y_1, \cdots, y_T \in \mathcal{P}$ whereas `TreeSwap` receives loss functions $\ell_1, \cdots, \ell_T : \mathcal{P} \to \mathbb{R}$.

To describe how the algorithms use the adversary's actions to produce the distributions $\mathbf{x}_t$, we need some additional ntation. The algorithms take as input parameters $H, L \in \mathbb{N}$ satisfying $H \geq 2$ and $H^{L-1} \leq T \leq H^L$. We index time steps $t \in [T]$ via base-H L-tuples: in particular, for $t \in [T]$, we let $t_1, \ldots, t_L \in [0 : H-1]$ be the base-H representation of $t-1$; we will write $t - 1 = (t_1 t_2 \cdots t_L)$. For all $0 \leq l \leq L$, for all $k \in [0 : H-1]^l$, let $\Gamma_k^{(l)} \subset [T]$ represent the interval of times t with prefix k. That is, $t \in \Gamma_k^{(l)}$ iff $t_i = k_i$ for all $i \in [1 : l]$. These intervals may be arranged to form an H-ary depth-L tree, where the children of $\Gamma_k^{(l)}$ are $\Gamma_{k0}^{(l+1)}, \Gamma_{k1}^{(l+1)}, \cdots, \Gamma_{k,H-1}^{(l+1)}$.[4]

Both `TreeCal` and `TreeSwap` operate by assigning an action $p_k^{(l)}$ to each node $\Gamma_k^{(l)}$ of the tree, except the root. At time t, both algorithms return the uniform distribution over the actions on the root-to-leaf-t path, namely $\mathbf{x}_t := \text{Unif}\left(\left\{p_{t_1}^{(1)}, p_{t_1 t_2}^{(2)}, \cdots, p_{t_1 t_2 \cdots t_L}^{(L)}\right\}\right)$ (see Figure 1). The algorithms differ in how the actions $p_k^{(l)}$ are chosen:

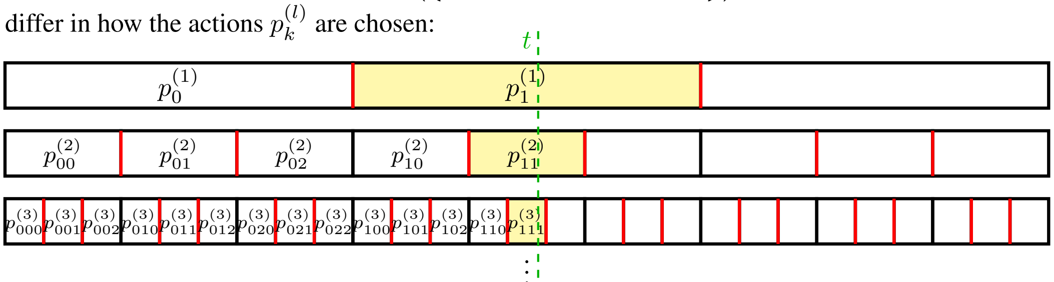

Figure 1: Visualization of the state of `TreeCal`/`TreeSwap` at time step t (about half-way through the algorithm). For $H = 3$, we depict the intervals Γ of the first three non-root levels of the tree ($l = 1, 2, 3$). Each rectangular node represents an interval, with sibling nodes separated by red lines. We represent the specific time step t via the vertical dashed green line. The yellow intervals it intersects at each level correspond to the nodes on the root-to-leaf-t path. Accordingly, $\mathbf{x}_t$ will be the uniform distribution over the labels p of these yellow intervals. We see that the algorithm has committed to the labels of all intervals that started at or before time t, and has yet to label the future intervals.

- `TreeCal` (Algorithm 1) assigns actions to nodes as follows. For all $1 \leq l \leq L$, $k \in [0 : H-1]^{l-1}$, $h \in [0 : H-1]$, at the start of $\Gamma_{kh}^{(l)}$, `TreeCal` sets $p_{kh}^{(l)}$ to be the average over all y_t that have been observed thus far in the parent interval $\Gamma_k^{(l-1)}$. That is,

$$p_{kh}^{(l)} = \frac{1}{hH^{L-l}} \sum_{i=0}^{h-1} \sum_{t \in \Gamma_{ki}^{(l)}} y_t \tag{4}$$

- The more general `TreeSwap` algorithm (Algorithm 2) also takes as a parameter an external regret-minimizing algorithm `Alg`, which operates with horizon of length H: we denote the resulting algorithm by `TreeSwap.Alg`. `TreeSwap.Alg` associates each internal node of the tree, $\Gamma_k^{(l-1)}$ (with $1 \leq l \leq L$), with an instance `Alg`, denoted $\texttt{Alg}_k^{(l-1)}$. The subroutine $\texttt{Alg}_k^{(l-1)}$ is responsible for choosing the actions $p_{k0}^{(l)}, p_{k1}^{(l)}, \cdots, p_{k(H-1)}^{(l)}$. It does so by responding to the average losses over each of its child intervals. In particular: at the end of each child interval $\Gamma_{kh}^{(l)}$, we pass $\texttt{Alg}_k^{(l-1)}$ the average loss over that interval. $\texttt{Alg}_k^{(l-1)}$ then outputs the action $p_{k(h+1)}^{(l)}$ assigned to the next child interval.

3.2 Main result

Theorem 3.1 upper bounds the calibration error of `TreeCal` with respect to the squared norm $\|\cdot\|^2$.

[4] We ignore the truncated branches that exist if $T < H^L$.

Theorem 3.1 (Main theorem)**.** *Let $\mathcal{P} \subset \mathbb{R}^d$ be a bounded convex set and $\|\cdot\|$ be an arbitrary norm. Then,* `TreeCal` *(Algorithm 1) guarantees that for an arbitrary sequence of outcomes $y_1, \dots, y_T \in \mathcal{P}$, the $\|\cdot\|^2$ calibration error of its predictions $\mathbf{x}_1, \dots, \mathbf{x}_T \in \Delta(\mathcal{P})$ is bounded as follows:*

$$\mathsf{Cal}_T^{\|\cdot\|^2}(\mathbf{x}_{1:T}, y_{1:T}) \le \epsilon T \quad \textit{for} \quad T \ge (\mathsf{diam}(\mathcal{P})/\sqrt{\epsilon})^{O(\mathsf{Rate}(\mathcal{P}, \|\cdot\|)/\epsilon)}$$

It is straightforward to derive from Theorem 3.1 via an application of Jensen's inequality an upper bound on the calibration error of `TreeCal` with respect to the (non-squared) norm $\|\cdot\|$, as stated in Theorem 1.1; see Corollary C.5. In Appendix E, we additionally consider a variant of `TreeCal` which plays *pure actions* in $\mathcal{P}$ (i.e., not distributions) by sampling from the distributions $\mathbf{x}_t$ for each $t \in [T]$. We show that the *pure calibration* error of this variant can be bounded by a similar quantity to that in Theorem 3.1.

3.3 Outline of the proof of Theorem 3.1

Step 1: Reduction from calibration error to swap regret. Let us choose a convex function $R : \mathcal{P} \to \mathbb{R}$ given $\mathcal{P}, \|\cdot\|$ as described in Section 2.4. The first step in the proof of Theorem 3.1 is to reduce the problem of minimizing (squared-norm) calibration error to that of minimizing full swap regret for an appropriate sequence of loss functions. In particular, for any sequence $\mathbf{x}_1, \dots, \mathbf{x}_T \in \Delta(\mathcal{P})$ and $y_1, \dots, y_T \in \mathcal{P}$, we have

$$\mathsf{Cal}_T^{\|\cdot\|^2}(\mathbf{x}_{1:T}, y_{1:T}) \le \mathsf{Cal}_T^{D_R}(\mathbf{x}_{1:T}, y_{1:T}) = \mathsf{FullSwapReg}_R(\mathbf{x}_{1:T}, \ell_{1:T}), \tag{5}$$

where $\ell_t : \mathcal{P} \to \mathbb{R}$ is the loss function given by $\ell_t(p) := D_R(y_t|p)$: the inequality uses strong convexity of R, and the subsequent equality uses Lemma 2.2.

Step 2: Equivalence with `TreeSwap`. Thus, it suffices to find an algorithm which minimizies the full swap regret quantity on the right-hand side of (5). Fortunately, the `TreeSwap` algorithm is known to do exactly this! (See Theorem C.1, from [DDFG24], for a formal statement for the swap regret bound of `TreeSwap`.) In order to apply the swap regret bound of Theorem C.1, we need to ensure that the `TreeCal` algorithm is an instantiation of `TreeSwap.Alg` for an appropriate choice of (a) the loss functions fed as input to `TreeSwap` and (b) the `Alg` subroutine. The loss functions have already been defined: given a sequence $y_1, \dots, y_T$, recall that we chose $\ell_t(p) := D_R(y_t|p)$. Moreover, we let the `Alg` subroutine be given by *Follow-the-Leader* (`FTL`), which simply chooses an action at each step minimizing the sum of losses up to the previous time step. The following lemma shows that `TreeSwap` with the losses ℓ_t and the `FTL` subroutine produces the same action distributions as `TreeCal`:

Lemma 3.2. *Let $\mathcal{P} \subset \mathbb{R}^d$ be a bounded convex set and let $R : \mathcal{P} \to \mathbb{R}$ be a convex function. For a sequence of loss functions $\ell_1, \cdots, \ell_H : \mathcal{P} \to \mathbb{R}$, define $\mathtt{FTL}_h(\ell_{1:H}) = \arg\min_{p \in \mathcal{P}} \sum_{s=1}^{h-1} \ell_s(p)$. For all sequences of outcomes $y_{1:T} \in \mathcal{P}^T$, the action distributions $\mathbf{x}_t$ produced by* `TreeCal` *on $y_{1:T}$ equal those produced by* `TreeSwap.FTL` *on loss functions $\ell_t(p) = D_R(y_t|p)$ for all t.*

The proof of Lemma 3.2 (given in full in the appendix) is a straightforward consequence of the fact that the Bregman divergence is a proper scoring rule: the action $p \in \mathcal{P}$ minimizing an average of Bregman divergences $D_R(y|p)$ is simply the average of the constituent points y (Lemma 2.1).

Step 3: Applying the swap regret bound of `TreeSwap` to `BTL`. Finally, we want to apply the main result of [DDFG24] (restated as Theorem C.1) to bound the full swap regret for the iterates $\mathbf{x}_{1:T}$ produced by `TreeSwap.Alg`, for an appropriate choice of `Alg`. The most natural way to do so would be to try to directly apply this result in the case when `Alg` = `FTL` (which corresponds to how we actually implement `TreeSwap`). However, applying this theorem requires an external regret bound on `FTL` for an arbitrary sequence of losses. While `FTL` is known to possess strong external regret bounds in some situations (e.g., when all the loss functions are strongly convex), the loss functions $p \mapsto D_R(y|p)$ are not necessarily even convex in p and so it is not a priori clear how to establish such bounds.

Instead, the main idea is to consider the "Be-The-Leader" algorithm `BTL`, which is the same as `FTL` but where actions are shifted ahead in time by 1 time step: in particular, the action chosen by `BTL` at time step h given a sequence $\ell_1, \ell_2, \dots, \ell_H : \mathcal{P} \to \mathbb{R}$ is $\mathtt{BTL}_h(\ell_{1:H}) = \mathtt{FTL}_{h+1}(\ell_{1:H}) =$

$\arg\min_{p\in\mathcal{P}}\sum_{s=1}^{h}\ell_s(p)$. BTL is not implementable since its action at time step h depends on the (unobserved) loss ℓ_h at that time step. However, since its regret is always non-positive (i.e., $\mathsf{ExtReg}_H(\texttt{BTL}) \le 0$ for any H), if we apply Theorem C.1 to the algorithm TreeSwap.BTL, we get that $\mathsf{FullSwapReg}_T(\texttt{TreeSwap.BTL}) \le \epsilon \cdot T$ as long as $T \ge H^{O(\rho/\epsilon)}$ for *any* choice of H (the arity parameter H used in TreeSwap). Using (5), this implies that the *calibration error* of the iterates produced by TreeSwap.BTL can also be bounded above by $\epsilon \cdot T$.

Of course, this result on its own is uninteresting (since BTL is unimplementable, as mentioned above). However, the key insight is that we can show that the actions chosen by TreeSwap.BTL are close to (as measured by the norm $\|\cdot\|$) those chosen by TreeSwap.FTL, which in turn is equivalent to TreeCal (Lemma 3.2). This closeness is an immediate consequence of the fact that the actions chosen by FTL for our loss functions $D_R(y_1|\cdot), D_R(y_2|\cdot), \dots$ are simply the empirical average of all actions $y_1, y_2, \dots \in \mathcal{P}$ of the adversary up to the previous time step.[5] In turn, we can use this closeness to show that the calibration error of TreeSwap.FTL is close to that of TreeSwap.BTL. This latter part of the argument becomes slightly tricky due to the possibility that different nodes of the tree might output the same action $p \in \mathcal{P}$; accordingly, we need to work with a *labeled* variant of the action set and bound the swap regret over this labeled variant; see Appendix C for further details.

4 Lower bound

To prove our calibration lower bound, we make use of the following swap regret lower bound.

Theorem 4.1 (Theorem 4.1 of [DFG+24]). *There is a sufficiently small constant $c_{4.1} > 0$ so that the following holds. Fix any $\epsilon > 0$. For any $d \in \mathbb{N}$, there is a subset $\mathcal{X} \subset [-1,1]^d$ so that the following holds for any $T \le \exp\left(c_{4.1}\min\{d^{1/14}, \epsilon^{-1/6}\}\right)$. There is an oblivious adversary producing a sequence $v_1, \dots, v_T$ with $\|v_t\|_1 \le 1$ and $\|v_t\|_\infty \le \max\{d^{-13/14}, \epsilon^{13/6}\}$ for all t, which satisfies the following property. For linear loss functions $\ell(x,v) = \langle v, x\rangle$ for vectors $v \in \mathbb{R}^d$ and $x \in \mathbb{R}^d$, any learning algorithm producing $\mathbf{x}_1, \dots, \mathbf{x}_T \in \Delta(\mathcal{X})$,*

$$\mathsf{FullSwapReg}_T(\mathbf{x}_{1:T}, \ell(\cdot, v_{1:T})) = \sup_{\pi:\mathcal{X}\to\mathcal{X}} \sum_{t=1}^{T}\sum_{p\in\mathcal{X}} \mathbf{x}_t(p)\cdot(\langle v_t, p\rangle - \langle v_t, \pi(p)\rangle) \ge \epsilon \cdot T.$$

We leverage the classic reduction from swap-regret minimization to calibration [FV98]: by producing calibrated predictions of the upcoming loss and best-responding to it, we can effectively minimize swap regret. This is formalized in the following lemma, proved in Appendix D.

Lemma 4.2. *Fix a set $\mathcal{P} \subset \mathbb{R}^d$, a norm $\|\cdot\|$, and write $D(p,p') := \|p-p'\|$. Suppose that, for some $\epsilon > 0, T \in \mathbb{N}$, there is an algorithm which chooses $\mathbf{x}_1, \dots, \mathbf{x}_T \in \Delta(\mathcal{P})$ and which ensures that for every oblivious adversary choosing $y_1, \dots, y_T \in \mathcal{P}$, we have $\mathsf{Cal}_T^D(\mathbf{x}_{1:T}, y_{1:T}) \le \epsilon \cdot T$. Then for every set $\mathcal{P}' \subset \mathbb{R}^d$, there is an algorithm which chooses $\mathbf{x}'_1, \dots, \mathbf{x}'_T \in \Delta(\mathcal{P}')$ and which ensures that for every oblivious adversary choosing $y_1, \dots, y_T \in \mathcal{P}$, we have*

$$\mathsf{FullSwapReg}_T(\mathbf{x}'_{1:T}, \ell(\cdot, y_{1:T})) \le \epsilon \cdot T \cdot \mathsf{diam}_{\|\cdot\|_\star}(\mathcal{P}').$$

Combining these two ideas, we demonstrate that an algorithm ϵ-calibrated predictions of outcomes on the simplex in $T \le \exp(\mathrm{poly}(1/\epsilon))$ rounds could be used in Lemma 4.2 to achieve a swap regret algorithm contradicting Theorem 4.1. This gives the following (proved in Appendix D).

Theorem 4.3. *There is a sufficiently small constant $c > 0$ so that the following holds. Write $D(p,p') = \|p-p'\|_1$, and fix any $\epsilon > 0, d \in \mathbb{N}$. Then for any $T \le \exp(c \cdot \min\{d^{1/14}, \epsilon^{-1/6}\})$, there is an oblivious adversary producing a sequence $y_1, \dots, y_T \in \Delta^d$ so that for any learning algorithm producing $\mathbf{x}_1, \dots, \mathbf{x}_T \in \Delta(\Delta^d)$, $\mathsf{Cal}_T^D(\mathbf{x}_{1:T}, y_{1:T}) \ge \epsilon \cdot T$.*

[5] An observant reader might note that this same argument also lets us provide bounds on the regret of FTL for these losses. One subtlety in the analysis is that we obtain better calibration bounds by bounding the distance between the predictions of FTL and BTL in the $\|\cdot\|$ norm rather than in the losses $D_R(y_t|\cdot)$, and so it is important that we directly analyze TreeSwap.BTL instead of TreeSwap.FTL (the latter causes us to pick up an extra factor related to the *smoothness* of R).

References

[ACRS25] Eshwar Ram Arunachaleswaran, Natalie Collina, Aaron Roth, and Mirah Shi. An elementary predictor obtaining distance to calibration. In *Proceedings of the 2025 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1366–1370. SIAM, 2025.

[AM11] Jacob Abernethy and Shie Mannor. Does an efficient calibrated forecasting strategy exist? In *Proceedings of the 24th Annual Conference on Learning Theory*, pages 809–812. JMLR Workshop and Conference Proceedings, 2011.

[BGHN23] Jarosław Błasiok, Parikshit Gopalan, Lunjia Hu, and Preetum Nakkiran. A unifying theory of distance from calibration. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, pages 1727–1740, 2023.

[BM07] Avrim Blum and Yishay Mansour. From external to internal regret. *Journal of Machine Learning Research*, 8(6), 2007.

[Daw82] A Philip Dawid. The well-calibrated bayesian. *Journal of the American statistical Association*, 77(379):605–610, 1982.

[DDF$^+$24] Yuval Dagan, Constantinos Daskalakis, Maxwell Fishelson, Noah Golowich, Robert Kleinberg, and Princewill Okoroafor. Breaking the $t^{2/3}$ barrier for sequential calibration. *arXiv preprint arXiv:2406.13668*, 2024.

[DDFG24] Yuval Dagan, Constantinos Daskalakis, Maxwell Fishelson, and Noah Golowich. From external to swap regret 2.0: An efficient reduction for large action spaces. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*, pages 1216–1222, 2024.

[DFG$^+$24] Constantinos Daskalakis, Gabriele Farina, Noah Golowich, Tuomas Sandholm, and Brian Hu Zhang. A lower bound on swap regret in extensive-form games. *arXiv preprint arXiv:2406.13116*, 2024.

[FH18] Dean P Foster and Sergiu Hart. Smooth calibration, leaky forecasts, finite recall, and nash dynamics. *Games and Economic Behavior*, 109:271–293, 2018.

[FKO$^+$25] Maxwell Fishelson, Robert Kleinberg, Princewill Okoroafor, Renato Paes Leme, Jon Schneider, and Yifeng Teng. Full swap regret and discretized calibration. *arXiv preprint arXiv:2502.09332*, 2025.

[FL99] Drew Fudenberg and David K Levine. An easier way to calibrate. *Games and economic behavior*, 29(1-2):131–137, 1999.

[Fos99] Dean P Foster. A proof of calibration via blackwell's approachability theorem. *Games and Economic Behavior*, 29(1-2):73–78, 1999.

[FV97] Dean P Foster and Rakesh V Vohra. Calibrated learning and correlated equilibrium. *Games and Economic Behavior*, 21(1-2):40–55, 1997.

[FV98] Dean P Foster and Rakesh V Vohra. Asymptotic calibration. *Biometrika*, 85(2):379–390, 1998.

[GJRR24] Sumegha Garg, Christopher Jung, Omer Reingold, and Aaron Roth. Oracle efficient online multicalibration and omniprediction. In *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2725–2792. SIAM, 2024.

[GPSW17] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1321–1330. PMLR, 06–11 Aug 2017.

[GSJ24] Khashayar Gatmiry, Jon Schneider, and Stefanie Jegelka. Computing optimal regularizers for online linear optimization. *arXiv preprint arXiv:2410.17336*, 2024.

[H$^+$16] Elad Hazan et al. Introduction to online convex optimization. *Foundations and Trends® in Optimization*, 2(3-4):157–325, 2016.

[Har22] Sergiu Hart. Calibrated forecasts: The minimax proof. *arXiv preprint arXiv:2209.05863*, 2022.

[HJKRR18] Úrsula Hébert-Johnson, Michael P. Kim, Omer Reingold, and Guy N. Rothblum. Multicalibration: Calibration for the (Computationally-identifiable) masses. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1939–1948. PMLR, 10–15 Jul 2018.

[HK12] Elad Hazan and Sham M Kakade. (weak) calibration is computationally hard. In *Conference on Learning Theory*, pages 3–1. JMLR Workshop and Conference Proceedings, 2012.

[HW24] Lunjia Hu and Yifan Wu. Predict to minimize swap regret for all payoff-bounded tasks. In *2024 IEEE 65th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 244–263. IEEE, 2024.

[KF08] Sham M Kakade and Dean P Foster. Deterministic calibration and nash equilibrium. *Journal of Computer and System Sciences*, 74(1):115–130, 2008.

[KLST23] Bobby Kleinberg, Renato Paes Leme, Jon Schneider, and Yifeng Teng. U-calibration: Forecasting for an unknown agent. In *The Thirty Sixth Annual Conference on Learning Theory*, pages 5143–5145. PMLR, 2023.

[LSS24] Haipeng Luo, Spandan Senapati, and Vatsal Sharan. Optimal multiclass u-calibration error and beyond. *arXiv preprint arXiv:2405.19374*, 2024.

[LSS25] Haipeng Luo, Spandan Senapati, and Vatsal Sharan. Simultaneous swap regret minimization via kl-calibration. *arXiv preprint arXiv:2502.16387*, 2025.

[MS10] Shie Mannor and Gilles Stoltz. A geometric proof of calibration. *Mathematics of Operations Research*, 35(4):721–727, 2010.

[MSA07] Shie Mannor, Jeff S Shamma, and Gürdal Arslan. Online calibrated forecasts: Memory efficiency versus universality for learning in games. *Machine Learning*, 67:77–115, 2007.

[NRRX23] Georgy Noarov, Ramya Ramalingam, Aaron Roth, and Stephan Xie. High-dimensional prediction for sequential decision making. *arXiv preprint arXiv:2310.17651*, 2023.

[Pen25] Binghui Peng. High dimensional online calibration in polynomial time. *arXiv preprint arXiv:2504.09096*, 2025.

[PR24] Binghui Peng and Aviad Rubinstein. Fast swap regret minimization and applications to approximate correlated equilibria. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*, pages 1223–1234, 2024.

[QV21] Mingda Qiao and Gregory Valiant. Stronger calibration lower bounds via sidestepping. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 456–466, 2021.

[QZ24] Mingda Qiao and Letian Zheng. On the distance from calibration in sequential prediction. In *The Thirty Seventh Annual Conference on Learning Theory*, pages 4307–4357. PMLR, 2024.

[RS24] Aaron Roth and Mirah Shi. Forecasting for swap regret for all downstream agents. In *Proceedings of the 25th ACM Conference on Economics and Computation*, pages 466–488, 2024.

[RST15] Alexander Rakhlin, Karthik Sridharan, and Ambuj Tewari. Sequential complexities and uniform martingale laws of large numbers. *Probability Theory and Related Fields*, 161(1-2):111–153, 2015.

[SS11] Shai Shalev-Shwartz. Online learning and online convex optimization. *Foundations and Trends in Machine Learning*, 4(2):107–194, 2011.

[SST11] Nati Srebro, Karthik Sridharan, and Ambuj Tewari. On the universality of online mirror descent. *Advances in neural information processing systems*, 24, 2011.

[ZME20] Shengjia Zhao, Tengyu Ma, and Stefano Ermon. Individual calibration with randomized forecasting. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 11387–11397. PMLR, 13–18 Jul 2020.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We prove all stated claims.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss limitations.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: We prove all theorems and lemmas.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [NA]

Justification: The paper does not include experiments.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [NA]

Justification: The paper does not include experiments requiring code.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [NA]

   Justification: The paper does not include experiments.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [NA]

   Justification: The paper does not include experiments.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [NA]

   Justification: The paper does not include experiments.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: The research conducted in the paper conforms, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: There is no societal impact of the work performed.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [NA]

    Justification: The paper does not use existing assets.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.

A Additional Related Work

There is a large range of other existing work on online (sequential) calibration [Daw82, FV97, FV98, QV21, DDF+24, Har22, Fos99, FL99, KF08, MSA07, MS10, AM11, HK12, FH18, LSS24, NRRX23, KLST23, GJRR24, QZ24, ACRS25]. We briefly survey some of these areas below.

Binary outcomes. For binary outcomes (i.e., one-dimensional calibration), classical results of [FV97, Fos99, BM07, AM11] demonstrate that it is possible to efficiently guarantee $O(T^{2/3})$ ℓ_1-calibration. The optimal possible rates for ℓ_1-calibration remain a major unsolved problem in online learning. Recently [QV21] improved over the naive lower bound of $\Omega(\sqrt{T})$ by demonstrating a lower bound of $\Omega(T^{0.528})$; this was further improved to $\Omega(T^{0.543})$ by [DDF+24], who also improved on the upper bound, demonstrating the existence of an algorithm with $O(T^{2/3-\epsilon})$ calibration for some constant $\epsilon > 0$.

Calibration and swap regret. The connection between calibration and swap regret has been acknowledged since the earliest works on swap regret. For example, the earliest algorithms for minimizing swap regret worked by best responding to online calibrated predictions [FV97] (later algorithms for swap regret minimization, such as [BM07] and [DDF+24] obtain better swap regret bounds by side-stepping the need to generate calibrated predictions). In the other direction, several works minimize calibration via relating it to a swap regret that can then be minimized [FKO+25, LSS25, AM11, Fos99].

Other forms of calibration. Due to the difficulty of minimizing (high-dimensional) calibration, there has been a line of work on designing forecasting algorithms that minimize weaker forms of calibration that recover some of the important guarantees of calibration (e.g., trustworthy-ness by a decision-maker). These include *distance from calibration* [BGHN23, QZ24, ACRS25], *omni-prediction error / U-calibration* [KLST23, LSS24, GJRR24], *calibration conditioned on downstream outcomes* [NRRX23], and *prediction for downstream swap regret* [RS24, HW24]. Other work focuses on minimizing notions of calibration designed to lead to specific classes of equilibria, e.g. weak calibration [HK12], deterministic calibration [KF08], and smooth calibration [FH18].

B Proofs of preliminary results

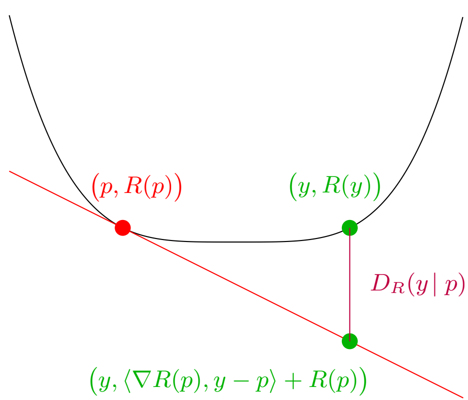

Figure 2: Geometric depiction of the Bregman divergence from p to y.

Proof of Lemma 2.1.

$$
\begin{aligned}
\mathbb{E}_{y\sim\mathbf{y}}[D_R(y|p)] &= \mathbb{E}_{y\sim\mathbf{y}}\left[R(y) - R(p) - \langle \nabla R(p), y-p\rangle\right] \\
&= \overline{R(y)} - R(p) - \langle \nabla R(p), \bar{y} - p\rangle \\
&= D_R(\bar{y}|p) + \overline{R(y)} - R(\bar{y})
\end{aligned}
$$

See Figure 3 for a visual proof. □

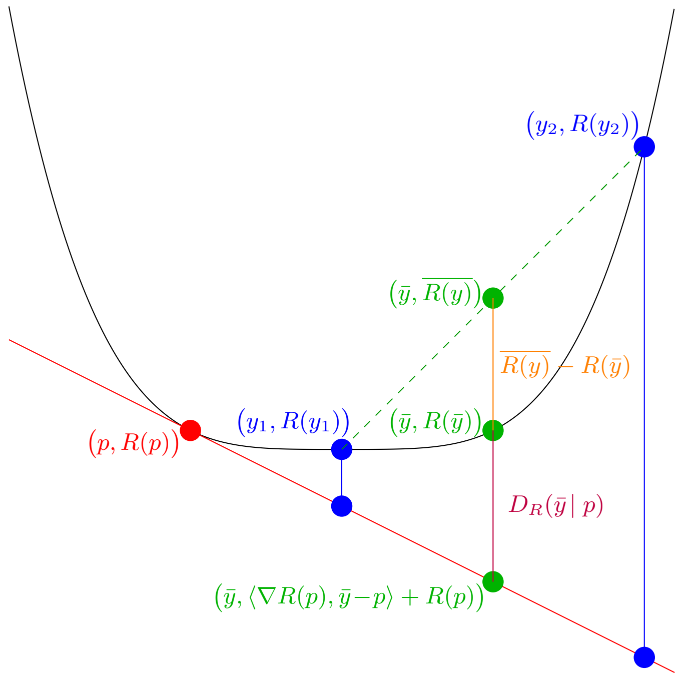

Figure 3: [Proof of Lemma 2.1] the average Bregman divergence (orange + purple) decomposes into the Jensen error (orange) and the Bregman divergence to the mean (purple). For example, when $R(p) = \|p\|_2^2$, $D_R(y|p) = \|y - p\|_2^2$ and we recover the bias-variance decomposition.

Proof of Lemma 2.2. Fix any $p \in \mathcal{P}$, and consider the quantity $\max_{p^* \in \mathcal{P}} \sum_t \mathbf{x}_t(p)(D_R(y_t|p) - D_R(y_t|p^*))$. By considering the distribution $\mathbf{y}$ that has weight $\mathbf{x}_t(p)/\sum_t \mathbf{x}_t(p)$ on y_t, Lemma 2.1 implies that this quantity is maximized when $p^* = \nu_p = (\sum_t \mathbf{x}_t(p) y_t)/(\sum_t \mathbf{x}_t(p))$. At this optimal value of p^*, this quantity can be rewritten as:

$$
\begin{aligned}
& \sum_t \mathbf{x}_t(p)(D_R(y_t|p) - D_R(y_t|\nu_p)) \\
= & \sum_t \mathbf{x}_t(p) \left[(R(y_t) - R(p) - \langle \nabla R(p), y_t - p \rangle) - (R(y_t) - R(\nu_p) - \langle \nabla R(\nu_p), y_t - \nu_p \rangle) \right] \\
= & \sum_t \mathbf{x}_t(p) \left[(R(\nu_p) - R(p) - \langle \nabla R(p), \nu_p - p \rangle) + \langle \nabla R(\nu_p) - \nabla R(p), y_t - \nu_p \rangle \right] \\
= & \sum_t \mathbf{x}_t(p) D_R(\nu_p|p) + \left\langle \nabla R(\nu_p) - \nabla R(p), \sum_t \mathbf{x}_t(p)(y_t - \nu_p) \right\rangle \\
= & \sum_t \mathbf{x}_t(p) D_R(\nu_p|p).
\end{aligned}
$$

(Here the last term vanishes since $\sum_t \mathbf{x}_t(p) y_t = \sum_t \mathbf{x}_t(p) \nu_p$). We therefore have that:

$$\begin{aligned}
\mathsf{FullSwapReg}_T(\mathbf{x}_{1:T}, \ell_{1:T}) &= \sup_{\pi:\mathcal{P}\to\mathcal{P}} \sum_{t=1}^{T}\sum_{p\in\mathcal{P}} \mathbf{x}_t(p)\cdot(\ell_t(p)-\ell_t(\pi(p))) \\
&= \sum_{p\in\mathcal{P}} \max_{p^*\in\mathcal{P}} \sum_{t=1}^{T} \mathbf{x}_t(p)\cdot(\ell_t(p)-\ell_t(p^*)) \\
&= \sum_{p\in\mathcal{P}} \max_{p^*\in\mathcal{P}} \sum_{t=1}^{T} \mathbf{x}_t(p)\cdot(D_R(y_t|p)-D_R(y_t|\nu_p)) \\
&= \sum_{p\in\mathcal{P}}\sum_{t} \mathbf{x}_t(p) D_R(\nu_p|p) \\
&= \mathsf{Cal}_T^{D_R}(\mathbf{x}_{1:T}, y_{1:T}).
\end{aligned}$$

□

Proof of Lemma 2.4. Note that if we define $\mathcal{L} = \{y \in \mathbb{R}^d \mid \|y\|_* \le 1\}$ to be the unit dual norm ball for some norm $\|\cdot\|$, then by duality the norm $\|\cdot\|_{\mathcal{L}^*}$ corresponding to $\mathcal{L}^*$ is simply the original norm $\|\cdot\|$. It therefore suffices to show that given a 1-strongly convex function R with bounded range ρ, it is possible to construct a 1-strongly convex function R' with bounded Bregman divergence $O(\rho)$ (and vice versa).

Assume $R(p)$ is 1-strongly convex and satisfies $\max_{p\in\mathcal{P}} R(p) - \min_{p\in\mathcal{P}} R(p) = \rho$. Define $R'(p) = 4R\left(\frac{p}{2}\right)$ (since $\mathcal{P}$ is centrally symmetric, $p/2$ is guaranteed to belong to $\mathcal{P}$). If R is 1-strongly convex, then $R(p/2)$ is $1/4$-strongly convex, and so $R'(p)$ is also 1-strongly convex. We claim the maximum Bregman divergence of R' is at most $O(\rho)$. To show this, we first argue that for any $z_1, z_2 \in \mathcal{P}$, $\left\langle \nabla R(\frac{z_1}{2}), z_2\right\rangle \le 2\rho$. To see this, note that since $R(p)$ is convex and has range bounded by ρ, we have that $\rho \ge R(p) - R(\frac{z_1}{2}) \ge \left\langle \nabla R(\frac{z_1}{2}), x - \frac{z_1}{2}\right\rangle$. If we set $p = \frac{z_1+z_2}{2}$, it then follows that $\left\langle \nabla R(\frac{z_1}{2}), z_2\right\rangle \le 2\rho$. Now, note that

$$\begin{aligned}
\max_{y,p\in\mathcal{P}} D_{R'}(y|p) &= R'(y) - R'(p) - \langle \nabla R'(p), y-p\rangle \\
&= R\left(\frac{y}{2}\right) - R\left(\frac{p}{2}\right) - \frac{1}{2}\left\langle \nabla R\left(\frac{p}{2}\right), y-p\right\rangle \\
&\le \left|R\left(\frac{y}{2}\right) - R\left(\frac{p}{2}\right)\right| + \left\langle \nabla R\left(\frac{p}{2}\right), \frac{y-p}{2}\right\rangle \le 3\rho.
\end{aligned}$$

Conversely, if $R(p)$ is 1-strongly convex and satisfies $\max_{y,p\in\mathcal{P}} D_R(y|p) \le \rho$, define $R'(p) = R(p) - \langle \nabla R(0), p\rangle - R(0)$ (i.e., subtracting a linear function to make zero a minimizer of $R'(p)$). Since R and R' differ by a linear function, R' is also 1-strongly convex. But also, note that $D_R(y|0) = R(y) - R(0) - \langle \nabla R(0), y\rangle = R'(y)$; since D_R is bounded in range by ρ, it follows that so is R'.

□

C Proof of Theorem 3.1

In this section, we prove Theorem 3.1. First, in Appendix C.1, we introduce a slightly stronger notion of calibration error and swap regret to deal with a technicality in the proof. We then give the proof of Theorem 3.1.

C.1 Labeled calibration and swap regret

Intuition. Recall that the `TreeCal` algorithm labels each interval $\Gamma_k^{(l)}$ of the tree with some action, $p_k^{(l)} \in \mathcal{P}$. At each time step t, the algorithm outputs the uniform distribution over all $p_k^{(l)}$

Algorithm 1 $\texttt{TreeCal}(\mathcal{P}, T, H, L)$

Require: Action set $\mathcal{P} \subset \mathbb{R}^d$, time horizon T, parameters H, L with $T \leq H^L$.

```
for 1 ≤ t ≤ T do
    Write the base-H representation of t − 1 as t = (h_1 ··· h_L), for h_1, ..., h_L ∈ [0 : H − 1].
    for 1 ≤ l ≤ L do
        Write k := (h_1 ··· h_{l−1}) ∈ [0 : H − 1]^{l−1}.
        if h_{l+1} = ··· = h_L = 0 or l = L then
            If h_l > 0, define ν^{(l)}_{k,h_l−1} := (1/H^{L−l}) · Σ_{s∈Γ^{(l)}_{k,h_l−1}} y_s.
            Define p^{(l)}_{k,h_l} := (1/h_l) Σ_{i=0}^{h_l−1} ν^{(l)}_{k,i} if h_l > 0, otherwise choose arbitrary p^{(l)}_{k,h_l} ∈ P.
        end if
    end for
    Output the uniform mixture x_t := Unif({p^{(1)}_{h_1}, ..., p^{(L)}_{h_1···h_L}}), and observe y_t.
end for
```

with $\Gamma_k^{(l)} \ni t$. When evaluating the calibration error, suppose that the actions $p_k^{(l)}$ are all distinct, for $l \in [L], k \in [0 : H-1]^{l-1}$ (as we discuss below, this case is in some sense the "worst case"). In this event, each action $p_k^{(l)}$ is compared to the average outcome over the interval $\Gamma_k^{(l)}$: $\bar{y}_k^{(l)} = \frac{1}{\left|\Gamma_k^{(l)}\right|} \sum_{t \in \Gamma_k^{(l)}} y_t$. Formally, this would give

$$\mathsf{Cal}_T^D(\mathbf{x}_{1:T}, y_{1:T}) = \sum_{l=1}^{L} \frac{H^{L-l}}{L} \sum_{k \in [H]^l} D\left(\bar{y}_k^{(l)}, p_k^{(l)}\right). \tag{6}$$

as each level l action is selected with $\frac{1}{L}$ mass for H^{L-l} rounds.

If it happened that two distinct intervals $\Gamma_{k_1}^{(l_1)}, \Gamma_{k_2}^{(l_2)}$ were assigned the same action $p = p_{k_1}^{(l_1)} = p_{k_1}^{(l_1)}$, then the calibration error would be *at most* the quantity on the right-hand side of (6) (by Jensen's inequality). In particular, rather than having to compare p to two potentially distinct quantities $D(\bar{y}_{k_1}^{(l_1)}, p), D(\bar{y}_{k_2}^{(l_2)}, p)$, the mass placed on p would be categorized under the same forecast and we would only compare p to an appropriately-weighted average of $\bar{y}_{k_1}^{(l_1)}$ and $\bar{y}_{k_2}^{(l_2)}$.

For technical reasons, it will turn out to be necessary to upper bound the "worst case quantity" on the right-hand side of (6) (and an analogous version for swap regret), even in the even that the actions $p_k^{(l)}$ are *not all distinct*. To streamline our notation, we introduce a generalization of these quantities which apply for arbitrary algorithms, which we call *labeled* calibration error and *labeled* swap regret.

Formal definitions. Given a convex set $\mathcal{P} \subset \mathbb{R}^d$, we define its *labeled* extension to be $\bar{\mathcal{P}} := \mathcal{P} \times \{0,1\}^\star$, i.e., elements of $\bar{\mathcal{P}}$ are tuples (p, σ), where $\sigma \in \{0,1\}^\star$ is a string that is said to *label* p. For a loss function $\ell : \mathcal{P} \to \mathbb{R}$, we extend its domain to $\bar{\mathcal{P}}$ in the natural way, i.e., $\ell((p, \sigma)) := \ell(p)$ for $(p, \sigma) \in \bar{\mathcal{P}}$. Given a sequence of distributions over the labeled extension, $\mathbf{x}_1, \ldots, \mathbf{x}_T \in \Delta(\bar{\mathcal{P}})$, and loss functions $\ell_1, \ldots, \ell_T : \mathcal{P} \to \mathbb{R}$, we define

$$\mathsf{FullSwapReg}_T(\mathbf{x}_{1:T}, \ell_{1:T}) := \sup_{\pi : \bar{\mathcal{P}} \to \bar{\mathcal{P}}} \sum_{t=1}^{T} \sum_{p \in \bar{\mathcal{P}}} \mathbf{x}_t(p) \cdot (\ell_t(p) - \ell_t(\pi(p))).$$

In words, the full swap regret of $\mathbf{x}_{1:T}$ with respect to $\ell_{1:T}$ is defined identically as in (2) except that the swap function π can now depend on the label σ. In particular, the labeled extension allows us to consider a more refined notion of swap regret where identical actions played in different rounds can be swapped (via π) to different alternatives as long as they have different labels.

In a similar manner we define the calibration error for a sequence of labeled distributions: given $\mathbf{x}_1, \ldots, \mathbf{x}_T \in \Delta(\bar{\mathcal{P}})$ and $y_1, \ldots, y_T \in \mathcal{P}$, we define

$$\mathsf{Cal}_T^D := \sum_{(p,\sigma) \in \bar{\mathcal{P}}} \left(\sum_{t=1}^{T} \mathbf{x}_t((p,\sigma)) \right) \cdot D(\nu_{(p,\sigma)}, p), \qquad \nu_{(p,\sigma)} := \frac{\sum_{t=1}^{T} \mathbf{x}_t((p,\sigma)) \cdot y_t}{\sum_{t=1}^{T} \mathbf{x}_t((p,\sigma))}.$$

The main result of [DDFG24] shows that the swap regret of `TreeSwap` is bounded, even when one labels the action produced at each node of the tree by the node of the tree. This labeled variant of `TreeSwap` is given in Algorithm 2. It functions exactly as discussed in Section 3.1, except that the distribution $\mathbf{x}_t$ output at time step t is in $\Delta(\bar{\mathcal{P}})$ instead of $\Delta(\mathcal{P})$. In particular, each $p_k^{(l)} \in \mathcal{P}$ in the support of $\mathbf{x}_t$ is labeled by the tuple $k \in [0 : H-1]^l$.[6]

Theorem C.1 (`TreeSwap`; Theorem 3.1 of [DDFG24])**.** *Suppose that $H, L \in \mathbb{N}$ satisfy $H \geq 2$ and $H^{L-1} \leq T \leq H^L$. For bounded convex action set $\mathcal{P} \subset \mathbb{R}^d$ and loss function set $\mathcal{L} \subset \{\ell : \mathcal{P} \to [0, b]\}$, let $\mathtt{Alg}_H : \mathcal{L}^H \to \bar{\mathcal{P}}^H$ be any algorithm. Then, the labeled `TreeSwap` algorithm (Algorithm 2) parametrized by $T, H, L, \mathcal{P}, \mathcal{L}, \mathtt{Alg}_H$ outputs labeled distributions $\mathbf{x}_1, \ldots, \mathbf{x}_T \in \Delta(\bar{\mathcal{P}})$ satisfying the following: for any sequence $\ell_1, \ldots, \ell_T \in \mathcal{L}$,*

$$\mathsf{FullSwapReg}_T(\mathbf{x}_{1:T}, \ell_{1:T}) \leq T \cdot \left(\frac{\mathsf{ExtReg}_H(\mathtt{Alg}_H)}{H} + \frac{3b}{L} \right).$$

Algorithm 2 $\mathtt{TreeSwap.Alg}(\mathcal{P}, \mathcal{L}, T, H, L)$, labeled variant (see Appendix C.1)

Require: Action set $\mathcal{P} \subset \mathbb{R}^d$, convex loss class $\mathcal{L} \subset (\mathcal{P} \to \mathbb{R})$, no-external regret algorithm `Alg`, time horizon T, parameters H, L with $T \leq H^L$.

1: For each sequence $h_1 \cdots h_{l-1} \in \bigcup_{l=1}^{L} [0 : H-1]^{l-1}$, initialize an instance of `Alg` with time horizon H, denoted $\mathtt{Alg}_{h_{1:l-1}}$.
2: **for** $1 \leq t \leq T$ **do**
3:     Write the base-H representation of $t-1$ as $t-1 = (h_1 \cdots h_L)$, for $h_1, \ldots, h_L \in [0 : H-1]$.
4:     **for** $1 \leq l \leq L$ **do**
5:         Write $k := (h_1 \cdots h_{l-1}) \in [0 : H-1]^{l-1}$.
6:         **if** $h_{l+1} = \cdots = h_L = 0$ or $l = L$ **then**
7:             If $h_l > 0$, define $\ell_{k,h_l-1}^{(l)} := \frac{1}{H^{L-l}} \cdot \sum_{s \in \Gamma_{k,h_l-1}^{(l)}} \ell_s \in \mathcal{L}$.
8:             Define $p_{k,h_l}^{(l)} = \mathtt{Alg}_{k,h_l+1}(\ell_{k,0:h_l-1}^{(l)}) \in \mathcal{P}$. ▷ *The h_lth action of $\mathtt{Alg}_k$ given the loss sequence $\ell_{k,1:h_l-1}^{(l)}$.*
9:         **end if**
10:     **end for**
11:     Output the uniform mixture $\mathbf{x}_t := \mathrm{Unif}(\{(p_{h_1}^{(1)}, h_1), \ldots, (p_{h_1 \cdots h_L}^{(L)}, h_{1:L})\}) \in \Delta(\bar{\mathcal{P}})$, and observe ℓ_t. ▷ *Each action $p_k^{(l)}$ is* labeled *by the sequence k (see Appendix C.1).*
12: **end for**

C.2 Proof of the main theorem

First, we recall some definitions from Section 3. For all $l \in [0 : L]$, for all $k \in [H]^l$, let $\Gamma_k^{(l)}$ represent the interval of times t with prefix k. That is, $t \in \Gamma_k^{(l)}$ iff $t_i = k_i$ for all $i \in [1 : l]$. These intervals form an H-ary depth-L tree, where the children of $\Gamma_k^{(l)}$ are $\Gamma_{k0}^{(l+1)}, \Gamma_{k1}^{(l+1)}, \cdots, \Gamma_{k(H-1)}^{(l+1)}$. In the calibration setting where the learner receives outcomes $y_{1:T}$, let $\nu_k^{(l)} = \frac{1}{|\Gamma_k^{(l)}|} \sum_{t \in \Gamma_k^{(l)}} y_t$ (as defined on Line 6 of Algorithm 1). In the swap regret setting where the learner receives loss functions $\ell_{1:T}$, let $\ell_k^{(l)} = \frac{1}{|\Gamma_k^{(l)}|} \sum_{t \in \Gamma_k^{(l)}} \ell_t$ (as defined in Line 7 of Algorithm 2).

Finally, recall that for an online learning algorithm `Alg` with time horizon H, we define its action at time step $h \in [H]$ given losses $\ell_1, \ldots, \ell_H : \mathcal{P} \to \mathbb{R}$ by $\mathtt{Alg}_h(\ell_1, \ldots, \ell_H)$. If $\mathtt{Alg}_h$ only depends on the first g losses, then we will write $\mathtt{Alg}_h(\ell_1, \ldots, \ell_g)$. In the proof of Theorem 3.1 we will consider two algorithms in particular; the first, Follow-The-Leader (`FTL`) is defined as follows: for

[6] Technically, the analysis of [DDFG24] does not analyse the labeled version, but the proof goes through as is – the only step where labeling changes any of the reasoning in the argument is in Eq. (8) of [DDFG24], where the upper bound as written in that equation holds even for the labeled version.

$\ell_1, \dots, \ell_{h-1} : \mathcal{P} \to \mathbb{R}$, we have

$$\mathtt{FTL}_h(\ell_1, \dots, \ell_{h-1}) = \operatorname*{argmin}_{p \in \mathcal{P}} \sum_{i=1}^{h-1} \ell_i(p).$$

The second algorithm we consider is the Be-The-Leader algorithm (BTL), which is defined as follows: for $\ell_1, \dots, \ell_h : \mathcal{P} \to \mathbb{R}$, we have

$$\mathtt{BTL}_h(\ell_1, \dots, \ell_h) = \operatorname*{argmin}_{p \in \mathcal{P}} \sum_{i=1}^{h} \ell_i(p).$$

Note that since $\mathtt{BTL}_h(\ell_{1:h})$ depends on the unobserved loss ℓ_h at time step h, it is unimplementable. Nevertheless, it will be useful in our analysis.

Next we prove Lemma 3.2, establishing the equivalence of `TreeCal` and `TreeSwap.FTL`. In fact, we establish the stronger claim, which immediately implies Lemma 3.2.

Lemma C.2. *Fix distributions $q_0, \dots, q_h \in \Delta(\mathcal{P})$, and define $\ell_h(p) := \mathbb{E}_{y \sim q_h}[D_R(y|p)]$. Then for each $h > 0$, $\mathtt{FTL}_h(\ell_0, \dots, \ell_{h-1}) = \frac{1}{h} \sum_{i=0}^{h-1} \mathbb{E}_{y \sim q_i}[y]$.*

Proof. The lemma is an immediate consequence of Lemma 2.1, noting that

$$\mathtt{FTL}_h(\ell_0, \dots, \ell_{h-1}) = \operatorname*{argmin}_{p \in \mathcal{P}} \sum_{i=0}^{h-1} \ell_i(p) = \operatorname*{argmin}_{p \in \mathcal{P}} \mathbb{E}_{i \sim [0:h-1], y \sim q_i}[D_R(y_i|p)] = \frac{1}{h} \sum_{i=0}^{h-1} \mathbb{E}_{y \sim q_i}[y]. \tag{7}$$

$\square$

Proof of Lemma 3.2. At time t, both `TreeCal` (Line 10 of Algorithm 1) and `TreeSwap.FTL` (Line 11 of Algorithm 2) select $\mathbf{x}_t = \mathrm{Unif}\left(\left\{p^{(1)}_{t_1}, p^{(2)}_{t_1 t_2}, \cdots, p^{(L)}_{t_1 t_2 \cdots t_L}\right\}\right)$. It remains to demonstrate that both algorithms assign actions $p^{(l)}_k$ to intervals $\Gamma^{(l)}_k$ identically. Fixing a choice of $l \in [L]$ and $k \in [0 : H-1]^{l-1}$, this is an immediate consequence of Lemma C.2 with $q_h = \mathrm{Unif}(\{y_t : t \in \Gamma^{(l)}_{k,h}\})$ and the fact that:

- In `TreeCal`, $p^{(l)}_{k,h} = \frac{1}{h} \sum_{i=0}^{h-1} \nu^{(l)}_{k,i}$ with $\nu^{(l)}_{k,i} = \mathbb{E}_{t \sim \mathrm{Unif}(\Gamma^{(l)}_{k,i})}[y_t]$;
- Whereas in `TreeSwap.FTL`, $p^{(l)}_{k,h} = \mathtt{FTL}_{h+1}(\ell^{(l)}_{k,0}, \dots, \ell^{(l)}_{k,h-1})$ with $\ell^{(l)}_{k,i} = \mathbb{E}_{t \sim \mathrm{Unif}(\Gamma^{(l)}_{k,i})}[D_R(y_t|\cdot)]$.

$\square$

We are now ready to prove Theorem 3.1.

Proof of Theorem 3.1. Fix any convex set $\mathcal{P}$ and a norm $\|\cdot\|$, and let $R : \mathcal{P} \to \mathbb{R}$ be chosen to be 1-strongly convex which has range $\rho > 0$. Lemma 3.2 gives that the actions $\mathbf{x}_1, \dots, \mathbf{x}_T \in \Delta(\mathcal{P})$ are identical to the actions played by `TreeSwap.FTL` with losses $\ell_t(p) = D_R(y_t|p)$ (Algorithm 2; we are ignoring the labels here). Thus, from here on, it suffices to bound the calibration error of the corresponding distributions $\mathbf{x}_1, \dots, \mathbf{x}_T$ of `TreeSwap.FTL`. The actions $p^{(l)}_{k,h}$ (for $l \in [L], k \in [0 : H-1]^{l-1}, h \in [0 : H-1]$) of `TreeSwap.FTL` satisfy $p^{(l)}_{k,h} = \mathtt{FTL}_{h+1}(\ell^{(l)}_{k,0}, \dots, \ell^{(l)}_{k,h-1})$.

Next, let $\tilde{p}^{(l)}_{k,h}$ denote the corresponding actions played by `TreeSwap.BTL`, i.e., $\tilde{p}^{(l)}_{k,h} = \mathtt{BTL}_{h+1}(\ell^{(l)}_{k,0}, \dots, \ell^{(l)}_{k,h})$. We let $\mathbf{x}_t \in \Delta(\bar{\mathcal{P}})$ denote the (labeled) distribution chosen by `TreeSwap.FTL` (Line 11 of Algorithm 2), and let $\tilde{\mathbf{x}}_t \in \Delta(\bar{\mathcal{P}})$ denote the corresponding distribution for `TreeSwap.BTL`. To be concrete, if $t - 1 = (h_1 \cdots h_L)$, then

$$\mathbf{x}_t = \mathrm{Unif}(\{(p^{(1)}_{h_1}, h_1), \dots, (p^{(L)}_{h_1 \cdots h_L}, h_{1:L})\}), \qquad \tilde{\mathbf{x}}_t = \mathrm{Unif}(\{(\tilde{p}^{(1)}_{h_1}, h_1), \dots, (\tilde{p}^{(L)}_{h_1 \cdots h_L}, h_{1:L})\}),. \tag{8}$$

We state the below claim, whose proof is deferred to the end of the section. (We remark that the primary purpose of introducing labeling is so that it is possible to establish Claim C.3.)

Claim C.3. *It holds that*

$$\mathsf{Cal}_T^{\|\cdot\|^2}(\mathbf{x}_{1:T}, y_{1:T}) - 2\mathsf{Cal}_T^{\|\cdot\|^2}(\tilde{\mathbf{x}}_{1:T}, y_{1:T}) \leq \frac{2 \cdot \mathsf{diam}(\mathcal{P})^2}{H^2} \cdot T. \tag{9}$$

The fact that BTL enjoys non-positive external regret (e.g., [SS11, Lemma 2.1] gives that for an arbitrary sequence of loss functions $\ell_t : \mathcal{P} \to \mathbb{R}$, the external regret of $\mathtt{BTL}_H$ satisfies $\mathsf{ExtReg}_H(\mathtt{BTL}_H) \leq 0$. Thus, by Theorem C.1, the swap regret of (the labeled version of) $\mathtt{TreeSwap}_T$ applied with $\mathtt{Alg}_H = \mathtt{BTL}_H$ may be bounded as follows: for any sequence of losses $\ell_1, \ldots, \ell_T : \mathcal{P} \to [0, \rho]$,

$$\mathsf{FullSwapReg}_T(\tilde{\mathbf{x}}_{1:T}, \ell_{1:T}) \leq T \cdot \frac{3\rho}{L}.$$

Using Lemma 2.2 and (9), we get that for an arbitrary sequence $y_1, \ldots, y_T \in \mathcal{P}$,

$$\begin{aligned}
\mathsf{Cal}_T^{\|\cdot\|^2}(\mathbf{x}_{1:T}, y_{1:T}) \leq& 2 \cdot \mathsf{Cal}_T^{\|\cdot\|^2}(\tilde{\mathbf{x}}_{1:T}, y_{1:T}) + \frac{2 \cdot \mathsf{diam}(\mathcal{P})^2}{H^2} \cdot T \\
\leq& 2 \cdot \mathsf{Cal}_T^{D_R}(\tilde{\mathbf{x}}_{1:T}, y_{1:T}) + \frac{2 \cdot \mathsf{diam}(\mathcal{P})^2}{H^2} \cdot T \\
=& 2 \cdot \mathsf{FullSwapReg}_T(\tilde{\mathbf{x}}_{1:T}, D_R(y_{1:T}|\cdot)) + \frac{2 \cdot \mathsf{diam}(\mathcal{P})^2}{H^2} \cdot T \\
\leq& \frac{6\rho \cdot T}{L} + \frac{2 \cdot \mathsf{diam}(\mathcal{P})^2 \cdot T}{H^2}.
\end{aligned}$$

Given any desired accuracy $\epsilon > 0$, choosing $L = 12\rho/\epsilon$ and $H = \mathsf{diam}(\mathcal{P})/\sqrt{\epsilon}$ gives that we can guarantee $\mathsf{Cal}_T^{\|\cdot\|^2}(\mathbf{x}_{1:T}, y_{1:T}) \leq \epsilon \cdot T$ as long as $T \geq H^L = (\mathsf{diam}(\mathcal{P})/\sqrt{\epsilon})^{12\rho/\epsilon^2}$. $\square$

Proof of Claim C.3. For each $t \in [T]$, we can write $t - 1 = h_1 h_2 \cdots h_L$ with $h_i \in [0 : H-1]$ for all $i \in [L]$, and $\mathbf{x}_t, \tilde{\mathbf{x}}_t$ are as given in (8). Let us write, for $(p, \sigma) \in \bar{\mathcal{P}}$,

$$\nu_{(p,\sigma)} := \frac{\sum_{t=1}^T \mathbf{x}_t((p,\sigma)) \cdot y_t}{\sum_{t=1}^T \mathbf{x}_t((p,\sigma))}, \qquad \tilde{\nu}_{(p,\sigma)} := \frac{\sum_{t=1}^T \tilde{\mathbf{x}}_t((p,\sigma)) \cdot y_t}{\sum_{t=1}^T \mathbf{x}_t((p,\sigma))},$$
$$\nu_\sigma := \frac{\sum_{p \in \mathcal{P}} \sum_{t=1}^T \mathbf{x}_t((p,\sigma)) \cdot y_t}{\sum_{p \in \mathcal{P}} \sum_{t=1}^T \mathbf{x}_t((p,\sigma))}. \tag{10}$$

Since each $p_{h_1 \cdots h_l}^{(l)}$ and each $\tilde{p}_{h_1 \cdots h_l}^{(l)}$ is labeled by $h_{1:l}$ in $\mathbf{x}_t$ and $\tilde{\mathbf{x}}_t$, respectively, it holds that for each σ of the form $\sigma = h_1 \cdots h_l$ (for some $l \in [L]$), there are unique $p, \tilde{p} \in \mathcal{P}$ so that $\nu_\sigma = \nu_{(p,\sigma)} = \nu_{(\tilde{p},\sigma)}$:

in particular, we have $p = p^{(l)}_{h_1\cdots h_l}, \tilde{p} = \tilde{p}^{(l)}_{h_1\cdots h_l}$. We can therefore bound

$$\begin{aligned}
&\mathsf{Cal}_T^{\|\cdot\|^2}(\mathbf{x}_{1:T}, y_{1:T}) - 2\mathsf{Cal}_T^{\|\cdot\|^2}(\tilde{\mathbf{x}}_{1:T}, y_{1:T}) \\
=&\sum_{l\in[L], h_{1:l}\in[0:H-1]^l} \left(\sum_{t=1}^{T} \mathbf{x}_t((p^{(l)}_{h_1\cdots h_l}, h_1\cdots h_l))\right) \cdot \|\nu_{h_1\cdots h_l} - p^{(l)}_{h_1\cdots h_l}\|^2 \\
&\qquad - 2\left(\sum_{t=1}^{T} \tilde{\mathbf{x}}_t((\tilde{p}^{(l)}_{h_1\cdots h_l}, h_1\cdots h_l))\right) \cdot \|\nu_{h_1\cdots h_l} - \tilde{p}^{(l)}_{h_1\cdots h_l}\|^2 \\
=&\sum_{l\in[L], h_{1:l}\in[0:H-1]^l} \frac{H^{L-l}}{L} \cdot \left(\left\|\nu_{h_1\cdots h_l} - p^{(l)}_{h_1\cdots h_l}\right\|^2 - 2\left\|\nu_{h_1\cdots h_l} - \tilde{p}^{(l)}_{h_1\cdots h_l}\right\|^2\right) \\
\leq& 2\sum_{l\in[L], h_{1:l}\in[0:H-1]^l} \frac{H^{L-l}}{L} \cdot \left\|p^{(l)}_{h_1\cdots h_l} - \tilde{p}^{(l)}_{h_1\cdots h_l}\right\|^2 \\
\leq& \frac{2}{L}\sum_{l=1}^{L} \sum_{h_{1:l-1}\in[0:H-1]^{l-1}} \mathsf{diam}(\mathcal{P})^2 \cdot H^{L-l} \\
\leq& \frac{2}{L}\sum_{l=1}^{L} \mathsf{diam}(\mathcal{P})^2 \cdot H^{L-1} \\
=& \frac{2T\mathsf{diam}(\mathcal{P})^2}{H},
\end{aligned} \tag{11}$$

where the second-to-last inequality uses that $\sum_{h_l=0}^{H-1} \left\|p^{(l)}_{h_1\cdots h_l} - \tilde{p}^{(l)}_{h_1\cdots h_l}\right\|^2 \leq \mathsf{diam}(\mathcal{P})^2$ for all choices of $h_1\cdots h_{l-1}$ (a consequence of Lemma C.4 and Lemma C.2).

□

Lemma C.4. *Fix any convex set $\mathcal{P} \subset \mathbb{R}^d$ and a convex function $R : \mathcal{P} \to \mathbb{R}$. Fix a sequence $y_1, \ldots, y_H \in \mathcal{P}$, and set*

$$p_h = \frac{1}{h-1}\sum_{i=1}^{h-1} y_i \;\forall h \in [H], h > 1, \qquad \tilde{p}_h = \frac{1}{h}\sum_{i=1}^{h} y_i \;\forall h \in [H],$$

as well as $p_1 \in \mathcal{P}$ arbitrarily. Then

$$\sum_{h=1}^{H} \|p_h - \tilde{p}_h\|^2 \leq 2 \cdot \mathsf{diam}(\mathcal{P})^2.$$

Proof. Note that

$$\tilde{p}_h - p_h = \frac{y_h}{h} - \frac{1}{h(h-1)}\sum_{i=1}^{h-1} y_i,$$

which implies that $\|\tilde{p}_h - p_h\|^2 \leq \frac{\pi^2}{6} \cdot \mathsf{diam}(\mathcal{P})^2 < 2\mathsf{diam}(\mathcal{P})^2$. □

Applying Cauchy-Schwarz, we get the following corollary,

Corollary C.5. *Let $\mathcal{P} \subset \mathbb{R}^d$ be a bounded convex set and $\|\cdot\|$ be an arbitrary norm. Then,* `TreeCal` *(Algorithm 1) guarantees that for an arbitrary sequence of outcomes $y_1, \ldots, y_T \in \mathcal{P}$, the $\|\cdot\|$ calibration error of its predictions $\mathbf{x}_1, \ldots, \mathbf{x}_T \in \Delta(\mathcal{P})$ is bounded $\mathsf{Cal}_T^{\|\cdot\|}(\mathbf{x}_{1:T}, y_{1:T}) \leq \epsilon T$ for $T \geq (\mathsf{diam}_{\|\cdot\|}(\mathcal{P})/\epsilon)^{O(\mathsf{Rate}(\mathcal{P},\|\cdot\|)/\epsilon^2)}$*

Proof. Using the fact that $\sum_{p\in\mathcal{P}}\sum_{t=1}^{T}\mathbf{x}_t(p)=1$ together with Jensen's inequality, we have

$$\begin{aligned}\frac{1}{T}\mathsf{Cal}_T^{\|\cdot\|}(\mathbf{x}_{1:T},y_{1:T}) &= \frac{1}{T}\sum_{p\in\mathcal{P}}\left(\sum_{t=1}^{T}\mathbf{x}_t(p)\right)\cdot\|\nu_p-p\| \\ &\le \sqrt{\frac{1}{T}\sum_{p\in\mathcal{P}}\left(\sum_{t=1}^{T}\mathbf{x}_t(p)\right)\cdot\|\nu_p-p\|^2} \\ &= \sqrt{\frac{1}{T}\mathsf{Cal}_T^{\|\cdot\|^2}(\mathbf{x}_{1:T},y_{1:T})}\le\epsilon\end{aligned}$$

for $T\ge(\mathsf{diam}(\mathcal{P})/\epsilon)^{O(\mathsf{Rate}(\mathcal{P},\|\cdot\|)/\epsilon^2)}$ by Theorem 3.1. Thus, $\mathsf{Cal}_T^{\|\cdot\|}(\mathbf{x}_{1:T},y_{1:T})\le\epsilon T$ for $T\ge(\mathsf{diam}(\mathcal{P})/\epsilon)^{O(\mathsf{Rate}(\mathcal{P},\|\cdot\|)/\epsilon^2)}$, incurring an additional factor of 2 in the exponent constant, as desired. □

Finally, for the setting of centrally symmetric $\mathcal{P}$, we can apply Lemma 2.4 to directly relate this regret bound to the optimal possible rate of an online linear optimization problem.

Corollary C.6. *Let $\mathcal{P}\subset\mathbb{R}^d$ be a bounded centrally symmetric convex set and $\|\cdot\|$ be an arbitrary norm. Then,* `TreeCal` *(Algorithm 1) guarantees that for an arbitrary sequence of outcomes $y_1,\dots,y_T\in\mathcal{P}$, the $\|\cdot\|$ calibration error of its predictions $\mathbf{x}_1,\dots,\mathbf{x}_T\in\Delta(\mathcal{P})$ is bounded $\mathsf{Cal}_T^{\|\cdot\|}(\mathbf{x}_{1:T},y_{1:T})\le\epsilon T$ for $T\ge(\mathsf{diam}_{\|\cdot\|}(\mathcal{P})/\epsilon)^{O(\mathsf{Rate}_{\mathsf{OLO}}(\mathcal{P},\|\cdot\|)/\epsilon^2)}$*

Proof. Follows immediately by applying Lemma 2.4 to Corollary C.5. □

D Proofs for Section 4

In this section, we prove lower bounds on high-dimensional calibration that tell us that in order to achieve calibration error at most $\epsilon\cdot T$, we need to take $T\gtrsim\exp(\mathrm{poly}(1/\epsilon))$. First, in Appendix D.1, we prove a lower bound for ℓ_1 calibration over the d-dimensional simplex, and then, in **??**, we prove a lower bound for ℓ_2 calibration over the unit d-dimensional Euclidean ball.

D.1 Lower bound on ℓ_1 calibration

First, we prove Theorem 4.3 which gives a lower bound on ℓ_1 calibration over the simplex $\mathcal{P}=\Delta^d$.

Proof of Lemma 4.2. Fix an algorithm `Alg` which ensures that $\mathsf{Cal}_T^{D}(\mathbf{x}_{1:T},y_{1:T})\le\epsilon\cdot T$ as in the statement of the lemma. We construct the following algorithm `Alg'`: it simulates `Alg`, but whenever `Alg` outputs the distribution $\mathbf{x}_t\in\Delta(\mathcal{P})$, `Alg'` chooses instead $\mathbf{x}'_t\in\Delta(\mathcal{P}')$, defined by

$$\mathbf{x}'_t(p') := \sum_{\substack{p\in\mathcal{P}:\\ p'=\mathrm{argmin}_{q\in\mathcal{P}'}\langle q,p\rangle}}\mathbf{x}_t(p).$$

To simplify notation, we define $\mathsf{BR}(p) := \mathrm{argmin}_{q \in \mathcal{P}'} \langle q, p \rangle$. It follows that, for any oblivious adversary choosing a (random) sequence $y_1, \ldots, y_T \in \mathcal{P}$,

$$\begin{aligned}
&\mathsf{FullSwapReg}_T(\mathbf{x}'_{1:T}, \ell(\cdot, y_{1:T})) \\
=& \sup_{\pi:\mathcal{P}' \to \mathcal{P}'} \sum_{p' \in \mathcal{P}} \sum_{t \in [T]} \mathbf{x}'_t(p') \cdot (\langle y_t, p' - \pi(p') \rangle) \\
=& \sup_{\pi:\mathcal{P}' \to \mathcal{P}'} \sum_{p \in \mathcal{P}} \sum_{t \in [T]} \mathbf{x}_t(p) \cdot (\langle y_t, \mathsf{BR}(p) - \pi(\mathsf{BR}(p)) \rangle) \\
=& \sup_{\pi:\mathcal{P}' \to \mathcal{P}'} \sum_{p \in \mathcal{P}} \left(\sum_{t \in [T]} \mathbf{x}_t(p) \right) \cdot (\langle \nu_p, \mathsf{BR}(p) - \pi(\mathsf{BR}(p)) \rangle) \\
=& \sup_{\pi:\mathcal{P}' \to \mathcal{P}'} \sum_{p \in \mathcal{P}} \left(\sum_{t \in [T]} \mathbf{x}_t(p) \right) \cdot (\langle \nu_p - p, \mathsf{BR}(p) - \pi(\mathsf{BR}(p)) \rangle + \langle p, \mathsf{BR}(p) - \pi(\mathsf{BR}(p)) \rangle) \\
\leq& \sup_{\pi:\mathcal{P}' \to \mathcal{P}'} \sum_{p \in \mathcal{P}} \left(\sum_{t \in [T]} \mathbf{x}_t(p) \right) \cdot (\|\nu_p - p\| \cdot \|\mathsf{BR}(p) - \pi(\mathsf{BR}(p))\|_\star) \\
\leq& \mathsf{diam}_{\|\cdot\|_\star}(\mathcal{P}') \cdot \mathsf{Cal}_T^D(\mathbf{x}_{1:T}, y_{1:T}),
\end{aligned}$$

where in the final inequality we have used the fact that $\|\mathsf{BR}(p) - \pi(\mathsf{BR}(p))\|_\star \leq \mathsf{diam}_{\|\cdot\|_\star}(\mathcal{P}')$. $\square$

For $p > 0$, $d \in \mathbb{N}$, write $\mathcal{B}_p^d := \{x \in \mathbb{R}^d \mid \|x\|_p \leq 1\}$ to denote the unit ℓ_p norm ball.

To map the lower bound Theorem 4.1 from the $\|\cdot\|_1$-norm unit ball $\mathcal{B}_1^d$ to the simplex and arrive at the desired contradiction using the above lemma, we use the following.

Lemma D.1. *Fix $d \in \mathbb{N}$, and write $D(x, y) := \|x - y\|_1$ for $x, y \in \mathbb{R}^d$. Suppose that there is an algorithm* `Alg` *for calibration over the domain $\mathcal{P} = \Delta^{2d+1}$ which produces $\mathbf{x}_{1:T}$ given the choices of an adversary $y_{1:T}$ achieving calibration error $\mathsf{Cal}_T^D(\mathbf{x}_{1:T}, y_{1:T}) \leq R(T)$, for $T \in \mathbb{N}$. Then there is an algorithm* `Alg`$'$ *for calibration over the domain $\mathcal{B}_1^d$ which produces $\mathbf{x}'_{1:T}$ given $y'_{1:T}$ achieving calibration error $\mathsf{Cal}_T^D(\mathbf{x}'_{1:T}, y'_{1:T}) \leq R(T)$.*

Proof of Lemma D.1. We define a mapping $\psi : \mathcal{B}_1^d \to \Delta^{2d+1}$ as follows: for $y \in \mathcal{B}_1^d \subset \mathbb{R}^d$, we define

$$\phi(y)_i = \begin{cases} [y_j]_+ & : i = 2j - 1, \; j \in [d] \\ [y_j]_- & : i = 2j, \; j \in [d] \\ 1 - \|y\| & : i = 2d + 1. \end{cases}$$

It is straightforward to see that ϕ has a left inverse ψ, defined as follows: for $z \in \Delta^{2d+1}$,

$$\psi(z)_i = z_{2i-1} - z_{2i}, \quad i \in [d],$$

so that $\psi \circ \phi(y) = y$ for all $y \in \mathbb{R}^d$.

We define the algorithm `Alg`$'$ as follows: given $y'_t \in \mathcal{B}_1^d \subset \mathbb{R}^d$, it defines $y_t \in \Delta^{2d+1}$ by $y_t = \phi(y'_t)$. `Alg`$'$ then feeds y_t into `Alg`, and if we denote the distribution output by `Alg` at time step t by $\mathbf{x}_t$, `Alg`$'$ then plays the push-forward measure $\mathbf{x}'_t := \psi \circ \mathbf{x}_t \in \Delta(\mathcal{B}_1^d)$.

Our bound on the calibration error of `Alg` gives

$$\mathsf{Cal}_T^D(\mathbf{x}_{1:T}, y_{1:T}) = \sum_{p \in \Delta^{2d+1}} \left(\sum_{t=1}^T \mathbf{x}_t(p) \right) \cdot \|\nu_p - p\|_1 \leq R(T),$$

where $\nu_p = \frac{\sum_{t=1}^T \mathbf{x}_t(p) \cdot y_t}{\sum_{t=1}^T \mathbf{x}_t(p)} \in \Delta^{2d+1}$. For $p' \in \mathcal{B}_1^d$, let us denote $\nu'_{p'} := \frac{\sum_{t=1}^T \mathbf{x}'_t(p') \cdot y'_t}{\sum_{t=1}^T \mathbf{x}'_t(p')} = \psi\left(\frac{\sum_{t=1}^T \mathbf{x}'_t(p') \cdot y_t}{\sum_{t=1}^T \mathbf{x}'_t(p')} \right)$, using linearity of ψ.

We may now bound the calibration error of $\texttt{Alg}'$ by

$$\begin{aligned}\mathsf{Cal}_T^D(\mathbf{x}'_{1:T}, y'_{1:T}) &= \sum_{p' \in \mathcal{B}_1^d} \left(\sum_{t=1}^T \mathbf{x}'_t(p') \right) \cdot \|\nu'_{p'} - p'\|_1 \\ &\leq \sum_{p \in \Delta^{2d+1}} \left(\sum_{t=1}^T \mathbf{x}_t(p) \right) \cdot \|\psi(\nu_p) - \psi(p)\|_1 \\ &\leq \mathsf{Cal}_T^D(\mathbf{x}_{1:T}, y_{1:T}).\end{aligned}$$

□

Proof of Theorem 4.3. Suppose to the contrary that there was an algorithm $\texttt{Alg}$ which bounded calibration error by ϵT for $T \leq \exp(c \cdot \min\{d^{1/14}, \epsilon^{-1/6}\})$. Then by Lemma D.1, for $d' = \lfloor (d-1)/2 \rfloor$ there is an algorithm $\texttt{Alg}'$ for calibration on the domain $\mathcal{B}_1^{d'} \subset \mathbb{R}^{d'}$ produces $\mathbf{x}'_{1:T}$ given $y'_{1:T}$ satisfying $\mathsf{Cal}_T^D(\mathbf{x}'_{1:T}, y'_{1:T}) \leq \epsilon \cdot T$ for any $T \leq \exp(c \cdot \min\{d^{1/14}, \epsilon^{-1/6}\})$.

We now apply Lemma 4.2 for $\mathcal{P} = \mathcal{B}_1^{d'} \subset \mathbb{R}^{d'}$, the norm given by the ℓ_1 norm $\|\cdot\|_1$, and $\mathcal{P}' := [-1,1]^{d'}$. Note that $\mathsf{diam}_{\|\cdot\|_\infty}(\mathcal{P}') = 1$. Then Lemma 4.2 ensures that there is an algorithm $\texttt{Alg}''$ which chooses $\mathbf{x}''_1, \ldots, \mathbf{x}''_T \in \Delta(\mathcal{P}')$ which ensures that for every oblivious adversary choosing $y''_1, \ldots, y''_T \in \mathcal{B}_1^{d'}$, we have $\mathsf{FullSwapReg}_T(\mathbf{x}''_{1:T}, \ell(\cdot, y''_{1:T})) \leq \epsilon \cdot T$.

But if $T \leq \exp(c_{4.1} \cdot \min\{(d')^{1/14}, \epsilon^{-1/6}\})$, we have a contradiction to Theorem 4.1, thus completing the proof of the theorem. □

D.2 Lower bound for ℓ_2 calibration

Next, we prove a lower bound for ℓ_2 calibration.

Theorem D.2. *There is a sufficiently small constant $c > 0$ so that the following holds. Write $D(p, p') = \|p - p'\|_2$ and fix any $\epsilon > 0$, $d \in \mathbb{N}$. Then for ant $T \leq \exp(c \cdot \min\{d^{1/14}, \epsilon^{-1/7}\})$, there is an oblivious adversary producing a sequence $y_1, \ldots, y_T \in \mathcal{B}_2^d$ so that for any learning algorithm producing $\mathbf{x}_1, \ldots, \mathbf{x}_T \in \Delta(\mathcal{B}_2^d)$, $\mathsf{Cal}_T^D(\mathbf{x}_{1:T}, y_{1:T}) \geq \epsilon \cdot T$.*

Proof. Fix $\epsilon > 0, d \in \mathbb{N}$, and write $\tilde{\epsilon} = \epsilon^{6/7}$. We may assume without loss of generality that $d \leq \tilde{\epsilon}^{-14/6}$, so that $\min\{d^{1/14}, \tilde{\epsilon}^{-1/6}\} = \min\{d^{1/14}, \epsilon^{-1/7}\} = d^{1/14}$: if this were not the case, we simply use the adversary resulting from $\tilde{\epsilon}^{-14/6}$ dimensions and project the forecaster's predictions down into this lower-dimensional subspace, which can only decrease calibration error. Now suppose to the contrary that there was an algorithm $\texttt{Alg}$ which bounded calibration error by ϵT for $T \leq \exp(c \cdot \min\{d^{1/14}, \epsilon^{-1/7}\}) = \exp(c \cdot d^{1/14})$. Then by Lemma 4.2 with $\mathcal{P} = \mathcal{B}_2^d$ and norm $\|\cdot\| = \|\cdot\|_2$, for any subset $\mathcal{P}' \subset \mathcal{B}_2^d$ we get that there is an algorithm which chooses $\mathbf{x}'_1, \ldots, \mathbf{x}'_T \in \Delta(\mathcal{P}')$ and which ensures that for every oblivious adversary choosing $y_1, \ldots, y_T \in \mathcal{B}_2^d$, we have

$$\mathsf{FullSwapReg}_T(\mathbf{x}'_{1:T}, (\langle \cdot, y_1 \rangle, \ldots, \langle \cdot, y_T \rangle)) \leq \epsilon \cdot T. \tag{12}$$

On the other hand, the oblivious adversary of Theorem 4.1 guarantees a subset $\mathcal{X} \subset [-1,1]^d \subset$ and an oblivious adversary producing a sequence $v_1, \ldots, v_T \in \mathbb{R}^d$ with $\|v_t\|_\infty \leq d^{-13/14}$ for all $t \in [T]$, so that

$$\mathsf{FullSwapReg}_T(\mathbf{x}_{1:T}, (\langle \cdot, v_1 \rangle, \ldots, \langle \cdot, v_T \rangle)) \geq \tilde{\epsilon} \cdot T \tag{13}$$

as long as $T \leq \exp(c_{4.1} \cdot d^{1/14})$. We have $\|v_t\|_2 \leq d^{1/2 - 13/14} = d^{-3/7}$ for all t, and scaling $\mathcal{X}$ down by a factor of $1/\sqrt{d}$ (i.e., letting $\mathcal{P}' = \mathcal{X}/\sqrt{d}$) and all vectors v_t up by a factor of $d^{3/7}$ (i.e., letting $v'_t = \sqrt{d} \cdot v_t$ ensures that any algorithm producing $\mathbf{x}'_1, \ldots, \mathbf{x}'_T \in \mathcal{P}'$ must still have full swap regret

$$\mathsf{FullSwapReg}_T(\mathbf{x}'_{1:T}, (\langle \cdot, v'_1 \rangle, \ldots, \langle \cdot, v'_T \rangle)) > \tilde{\epsilon} \cdot T \cdot d^{-1/14} \geq \tilde{\epsilon}^{7/6} \cdot T = \epsilon \cdot T,$$

but now ensures that $\mathcal{P}' \subset \mathcal{B}_2^d$ and that $v'_t \in \mathcal{B}_2^d$ for all t. By taking $c = c_{4.1}$, this contradicts (12). □

E Pure calibration and pure full swap regret

E.1 Pure calibration

In certain settings of calibration, the learner is required to randomly select a pure forecast $p_t \in \mathcal{P}$ rather than a distribution $\mathbf{x}_t \in \Delta(\mathcal{P})$. In these settings, the above definition of calibration is instead referred to as "pseudo-calibration". Here, we stick to calling the above calibration, as we believe it to be the more natural definition, and instead refer to this alternative setting as "pure-calibration". The learning task changes as follows.

At each time step $t \in [T]$:

- The learning algorithm chooses a distribution $\mathbf{x}_t \in \Delta(\mathcal{P})$.
- The adversary observes $\mathbf{x}_t$ and chooses an *outcome* $y_t \in \mathcal{P}$.
- The learner samples $p_t \sim \mathbf{x}_t$.

We adjust the definitions of the "pure average outcome" and "pure-calibration" accordingly:

$$\dot{\nu}_p := \frac{\sum_{t=1}^{T} \mathbb{1}[p_t = p] \cdot y_t}{\sum_{t=1}^{T} \mathbb{1}[p_t = p]}, \qquad \mathsf{PureCal}_T^D(p_{1:T}, y_{1:T}) := \sum_{p \in \mathcal{P}} \left(\sum_{t=1}^{T} \mathbb{1}[p_t = p] \right) \cdot D(\dot{\nu}_p, p)$$

Algorithm 3 $\texttt{SampleTreeCal}(\mathcal{P}, T, H, L, S)$

Require: Action set $\mathcal{P} \subset \mathbb{R}^d$, time horizon T, repetition parameter S parameters H, L with $T/S \leq H^L$.

Instantiate an instance $\texttt{TreeCal}(\mathcal{P}, T/S, H, L)$.
for $1 \leq i \leq T/S$ **do**
  Let $\mathbf{x}_i \in \Delta(\mathcal{P})$ denote the prediction of $\texttt{TreeCal}$ at step i.
  for $1 \leq j \leq S$ **do**
    Sample $p_{S(i-1)+j} \sim \mathbf{x}_i$, and observe outcome $y_{S(i-1)+j}$.
  end for
  Feed the outcome $\bar{y}_i := \frac{1}{S} \sum_{j=1}^{S} y_{S(i-1)+j}$ to $\texttt{TreeCal}$.
end for

To obtain a bound on the (expected) pure calibration error, we use a slight modification of $\texttt{TreeCal}$, namely $\texttt{SampleTreeCal}$ (Algorithm 3). It functions identically to $\texttt{TreeCal}$ except that for each time step t of $\texttt{TreeCal}$, it samples S actions from $\mathbf{x}_t$ on each of S contiguous time steps. (Hence, $\texttt{TreeCal}$ is used with time horizon T/S.) At a high level, we will use an appropriate concentration inequality to show that the calibration upper bound of Theorem 3.1 implies a *pure calibration* upper bound for $\texttt{SampleTreeCal}$.

Theorem E.1 (Pure calibration error). *Let $\mathcal{P} \subset \mathbb{R}^d$ be a bounded convex set and $\|\cdot\|$ be an arbitrary norm with unit dual ball $\mathcal{L} := \{f \in \mathbb{R}^d \mid \|f\|_\star \leq 1\}$. Then, $\texttt{SampleTreeCal}$ (Algorithm 3, with an appropriate choice of parameters H, L, S) guarantees that for an arbitrary sequence of outcomes $y_1, \ldots, y_T \in \mathcal{P}$, the $\|\cdot\|^2$ calibration error of its predictions $\mathbf{x}_1, \ldots, \mathbf{x}_T \in \Delta(\mathcal{P})$ is bounded as follows:*

$$\mathbb{E}[\mathsf{PureCal}_T^{\|\cdot\|}(p_{1:T}, y_{1:T})] \leq \epsilon T, \quad \textit{for} \quad T \geq \mathsf{Rate}(\mathcal{L}, \|\cdot\|_\star) \cdot (\mathsf{diam}_{\|\cdot\|}(\mathcal{P})/\epsilon)^{O(\mathsf{Rate}(\mathcal{P}, \|\cdot\|)/\epsilon^2)}.$$

Proof. The proof uses Theorem 3.1 together with an appropriate concentration inequality, and closely follows that of [Pen25, Lemma 3.4].

Fix any $1 \leq i \leq T/S$ and $1 \leq j \leq S$, and let $\mathscr{F}_{S(i-1)+j}$ denote the σ-algebra generated by $y_1, \ldots, y_{S(i-1)+j+1}$ and $p_1, \ldots, p_{S(i-1)+j}$; since $\texttt{TreeCal}$ is deterministic, it follows that $\mathbf{x}_1, \ldots, \mathbf{x}_i \in \Delta(\mathcal{P})$ are $\mathscr{F}_i$-measurable. For any $1 \leq j \leq S$, we have that, for any $p \in \mathrm{supp}(\mathbf{x}_i)$,

$$\mathbb{E}\left[(p - y_{S(i-1)+j}) \cdot \mathbb{1}[p_{S(i-1)+j} = p] \mid \mathscr{F}_{S(i-1)+j-1}\right] = (p - y_{S(i-1)+j}) \cdot \mathbf{x}_i(p).$$

Fixing any $i \in [T/S]$, By Lemma E.2 applied to the sequence $M_{S(i-1)+j} = (p - y_{S(i-1)+j}) \cdot \mathbb{1}[p_{S(i-1)+j} = p]$, for $1 \le j \le S$ (and the filtration $\mathscr{F}_{S(i-1)+j}$), we see that

$$\mathbb{E}\left[\left\|\sum_{j=1}^{S}(p - y_{S(i-1)+j}) \cdot \mathbb{1}[p_{S(i-1)+j} = p] - \sum_{j=1}^{S}(p - y_{S(i-1)+j}) \cdot \mathbf{x}_i(p)\right\|\right] \le \mathsf{diam}_{\|\cdot\|}(\mathcal{P}) \cdot \sqrt{8S \cdot \mathsf{Rate}(\mathcal{L}, \|\cdot\|_\star)}.$$

It follows by summing over the L values of $p \in \mathrm{supp}(\mathbf{x}_i)$ that

$$\begin{aligned}&\mathbb{E}\left[\sum_{p\in\mathcal{P}}\left\|\sum_{j=1}^{S}(p - y_{S(i-1)+j}) \cdot \mathbb{1}[p_{S(i-1)+j} = p] - \sum_{j=1}^{S}(p - y_{S(i-1)+j}) \cdot \mathbf{x}_i(p)\right\|\right] \\ \le& L \cdot \mathsf{diam}_{\|\cdot\|}(\mathcal{P}) \cdot \sqrt{8S \cdot \mathsf{Rate}(\mathcal{L}, \|\cdot\|_\star)} \le \epsilon \cdot S, \end{aligned} \tag{14}$$

as long as $S \ge \frac{8\cdot\mathsf{Rate}(\mathcal{L},\|\cdot\|_\star)\cdot\mathsf{diam}_{\|\cdot\|}(\mathcal{P})^2\cdot L^2}{\epsilon^2}$.

The guarantee of Corollary C.5 gives that, as long as $T/S \ge (\mathsf{diam}_{\|\cdot\|}(\mathcal{P})/\epsilon)^{O(\mathsf{Rate}(\mathcal{P},\|\cdot\|)/\epsilon^2)}$, then

$$\begin{aligned}\mathsf{Cal}_T^{\|\cdot\|}(\mathbf{x}_{1:T/S}, \bar{y}_{1:T/S}) &= \sum_{p\in\mathcal{P}}\left\|\sum_{i=1}^{T/S}\mathbf{x}_i(p)\cdot(p - \bar{y}_i)\right\| \\ &= \sum_{p\in\mathcal{P}}\left\|\sum_{i=1}^{T/S}\frac{1}{S}\sum_{j=1}^{S}\mathbf{x}_i(p)\cdot(p - y_{S(i-1)+j})\right\| \le \frac{\epsilon T}{S}. \end{aligned} \tag{15}$$

By combining Equations (14) and (15), it follows that for an arbitrary adaptive adversary who chooses a sequence $y_1, \ldots, y_T \in \mathcal{P}$,

$$\begin{aligned}&\mathbb{E}\left[\mathsf{PureCal}_T^{\|\cdot\|}(p_{1:T}, y_{1:T})\right] \\ =&\mathbb{E}\left[\sum_{p\in\mathcal{P}}\left\|\sum_{t=1}^{T}(p - y_t)\cdot\mathbb{1}[p_t = p]\right\|\right] \\ \le&\mathbb{E}\left[\sum_{p\in\mathcal{P}}\left\|\sum_{i=1}^{T/S}\sum_{j=1}^{S}(p - y_{S(i-1)+j})\cdot\mathbf{x}_i(p)\right\| + \sum_{i=1}^{T/S}\left\|\sum_{j=1}^{S}\left((p - y_{S(i-1)+j})\cdot(\mathbb{1}[p_{S(i-1)+j} = p] - \mathbf{x}_i(p))\right)\right\|\right] \\ \le&2\epsilon T.\end{aligned}$$

The result follows by rescaling ϵ and our choice of $L = O(\mathsf{Rate}(\mathcal{P}, \|\cdot\|)/\epsilon^2)$. □

As example applications of Theorem E.1:

- When $\|\cdot\|$ is the ℓ_1 norm and $\mathcal{P}$ is the simplex, we have $\mathsf{diam}_{\|\cdot\|}(\mathcal{P}) = 1$, $\mathcal{L} = \{f \in \mathbb{R}^d \mid \|f\|_\infty \le 1\}$ satisfies $\mathsf{Rate}(\mathcal{L}, \|\cdot\|_\star) \le d$ (as we can take the function $R(x) = \|x\|_2^2$), which gives that for $T \ge d^{O(1/\epsilon^2)}$, we can have $\mathbb{E}[\mathsf{PureCal}_T^{\|\cdot\|_1}] \le \epsilon T$. This result recovers the main upper bound of [Pen25] (Theorem 1.1 therein).
- When $\|\cdot\|$ is the ℓ_2 norm and $\mathcal{P}$ is the unit ℓ_2 ball, we have $\mathsf{diam}_{\|\cdot\|}(\mathcal{P}) = 1$, $\mathcal{L} = \{f \in \mathbb{R}^d \mid \|f\|_2 \le 1\}$ satisfies $\mathsf{Rate}(\mathcal{L}, \|\cdot\|_\star) \le 1$ (as we can take the function $R(x) = \|x\|_2^2$), which gives that for $T \ge \exp(O(1/\epsilon^2))$, we can have $\mathbb{E}[\mathsf{PureCal}_T^{\|\cdot\|_1}] \le \epsilon T$.

E.2 Sequential law of large numbers

Fix a convex set $\mathcal{P} \subset \mathbb{R}^d$ and a norm $\|\cdot\|$ on $\mathbb{R}^d$. We define

$$\mathcal{R}_n(\mathcal{P}, \|\cdot\|) := \sup_{\mathbf{p}} \mathbb{E}_\epsilon\left[\left\|\frac{1}{n}\sum_{i=1}^{n}\epsilon_i\mathbf{p}_i(\epsilon)\right\|\right],$$

where the supremum is over all sequences of mappings $\mathbf{p}_1, \ldots, \mathbf{p}_n$, where $\mathbf{p}_i : \{-1, 1\}^{i-1} \to \mathcal{P}$, and the expectation is over an i.i.d. sequence of Rademacher random variables $\epsilon = (\epsilon_1, \ldots, \epsilon_n)$, $\epsilon_i \sim \mathrm{Unif}(\{\pm 1\})$. The below lemma (essentially contained in [RST15]) establishes a martingale law of large numbers for $\mathcal{P}$-valued martingales, in terms of geometric properties of $\mathcal{P}$ and $\|\cdot\|$.

Lemma E.2 ([RST15])**.** *Consider a convex set $\mathcal{P} \subset \mathbb{R}^d$ a norm $\|\cdot\|$ on $\mathbb{R}^d$, and let $M_1, \ldots, M_n$ denote a sequence of random variables adapted to a filtration $(\mathscr{F}_i)_{i \in [n]}$. Let $\mathcal{L} = \{f \mid \|f\|_\star \le 1\}$ be the unit ball of the dual norm $\|\cdot\|$. Then*

$$\mathbb{E}\left[\left\|\sum_{i=1}^n M_i - \mathbb{E}[M_i \mid \mathscr{F}_{i-1}]\right\|\right] \le \mathsf{diam}_{\|\cdot\|}(\mathcal{P}) \cdot \sqrt{8n \cdot \mathsf{Rate}(\mathcal{L}, \|\cdot\|_\star)}.$$

Proof. By applying an appropriate translation to $\mathcal{P}$, we can assume that $\mathcal{P}$ contains the origin. We apply Theorem 2 of [RST15] with the domain $\mathcal{Z}$ equal to $\mathcal{P}$ and the function class $\mathcal{F}$ equal to the class of mappings $\{z \mapsto \langle z, f\rangle \;:\; \|f\|_\star \le 1\}$ indexed by unit-dual norm linear functions on $\mathcal{Z}$. The theorem implies that

$$\begin{aligned}\mathbb{E}\left[\frac{1}{n}\left\|\sum_{i=1}^n M_i - \mathbb{E}[M_i \mid \mathscr{F}_{i-1}]\right\|\right] \le& 2 \cdot \sup_{\mathbf{p}} \mathbb{E}_\epsilon\left[\sup_{\|f\|_\star \le 1} \frac{1}{n}\left\langle \sum_{i=1}^n \epsilon_i \mathbf{p}_i(\epsilon), f\right\rangle\right] \\ =& 2 \cdot \mathcal{R}_n(\mathcal{P}, \|\cdot\|)).\end{aligned}$$

Write $\mathcal{L} = \{f \in \mathbb{R}^d \;:\; \|f\|_\star \le 1\}$ denote the unit ball for the dual norm $\|\cdot\|_\star$. Proposition 16 of [RST15] gives that, if there is a function $R : \mathcal{L} \to \mathbb{R}$ which is 1-strongly convex with respect to $\|\cdot\|_\star$ and which has range ρ, then $\mathcal{R}_n(\mathcal{P}, \|\cdot\|) \le \sqrt{\frac{2\rho}{n}} \cdot \mathsf{diam}_{\|\cdot\|}(\mathcal{P})$. In particular, $\mathcal{R}_n(\mathcal{P}, \|\cdot\|) \le \mathsf{diam}_{\|\cdot\|}(\mathcal{P}) \cdot \sqrt{\frac{2\mathsf{Rate}(\mathcal{L}, \|\cdot\|_\star)}{n}}$. $\square$