Near-Optimal Algorithms for Omniprediction

Princewill Okoroafor* Robert Kleinberg Michael P. Kim†‡

Abstract

Omnipredictors are simple prediction functions that encode loss-minimizing predictions with respect to a hypothesis class $\mathcal{H}$, simultaneously for every loss function within a class of losses $\mathcal{L}$. In this work, we give near-optimal learning algorithms for omniprediction, in both the online and offline settings. To begin, we give an oracle-efficient online learning algorithm that acheives $(\mathcal{L}, \mathcal{H})$-omniprediction with $\tilde{O}(\sqrt{T \log |\mathcal{H}|})$ regret for any class of Lipschitz loss functions $\mathcal{L} \subseteq \mathcal{L}_{\text{Lip}}$. Quite surprisingly, this regret bound matches the optimal regret for *minimization of a single loss function* (up to a $\sqrt{\log(T)}$ factor). Given this online algorithm, we develop an online-to-offline conversion that achieves near-optimal complexity across a number of measures. In particular, for all bounded loss functions within the class of Bounded Variation losses $\mathcal{L}_{\text{BV}}$ (which include all convex, all Lipschitz, and all proper losses) and any (possibly-infinite) $\mathcal{H}$, we obtain an offline learning algorithm that, leveraging an (offline) ERM oracle and m samples from $\mathcal{D}$, returns an efficient $(\mathcal{L}_{\text{BV}}, \mathcal{H}, \varepsilon(m))$-omnipredictor for $\varepsilon(m)$ scaling near-linearly in the Rademacher complexity of $\text{Th} \circ \mathcal{H}$.

*PO was in part supported by a Linkedin-Cornell Fellowship Grant.

†MPK: This research was supported by a gift to the LinkedIn-Cornell Bowers CIS Strategic Partnership.

‡Authors listed in reverse-alphabetical order.

Contents

1 Introduction

In recent years, the research community has investigated learning frameworks beyond the traditional objective of loss minimization. Amidst research addressing concerns of fairness and robustness, Gopalan et al. (2022) introduced a powerful notion of robust learning, called *omniprediction*. An omnipredictor is a simple prediction function for binary outcomes that encodes loss-minimizing predictions *simultaneously* for every loss within a broad class of loss functions $\mathcal{L}$. In contrast to the conventional wisdom—that exploring different loss functions necessitates training different loss minimizers—omnipredictors allow a decision-maker to learn a single model p (an omnipredictor) and subsquently decide which loss function ℓ is appropriate for their setting, without retraining p.

More technically, omniprediction is a parameterized guarantee, based on a class of loss functions $\mathcal{L}$ as well as a hypothesis class $\mathcal{H}$, and requires loss minimization for every $\ell \in \mathcal{L}$ with respect to $\mathcal{H}$.

Definition. A prediction function $p : \mathcal{X} \to [0, 1]$ is an $(\mathcal{L}, \mathcal{H}, \varepsilon)$-omnipredictor if for every $\ell \in \mathcal{L}$

$$\mathbf{E}[\ell(k_\ell \circ p(x), y)] \leq \min_{h \in \mathcal{H}} \mathbf{E}[\ell(h(x), y)] + \varepsilon.$$

That is, for every loss ℓ within the collection, $k_\ell \circ p$ is a loss minimizer for ℓ. Here, k_ℓ is a data-free post-processing of the predictions given by p that "type-checks" the outputs given by p to match ℓ, specifically: $k_\ell(v) \triangleq \arg\min_{a \in [0,1]} \mathbf{E}_{y \sim \mathrm{Ber}(v)}[\ell(a, y)]$. Such a post-processing of p is necessary for the omnipredictor definition, as different losses expect different "types" of optimal predictions.[1]

On its face, the guarantee of omniprediction significantly strengthens that of loss minimization with respect to a single loss function, especially when faced with a large, diverse class of losses. A natural question, facing such a strong definition, is whether omnipredictors exist. With some thought, it's not hard to see that the optimal predictor $p^*(x) = \mathbf{Pr}[y = 1|x]$ is an omnipredictor for every $\mathcal{L}$ and $\mathcal{H}$. That is, if we know the true conditional distribution of outcomes given covariates, we can make (statistically) optimal predictions according to every loss function. In most learning settings, however, the optimal predictor p^* is unattainable, both statistically and computationally.

A more delicate question is whether *efficient* omnipredictors exist, with complexity scaling comparably to the hypotheses the omnipredictor is competing against. Moreover, if such omnipredictors exist, can we learn them from a finite sample of data, using reasonable amounts of computation? The original work introducing omniprediction showed the sweeping result that efficient[2] omniprediction is possible for the class of all convex loss functions.

Theorem (Efficient Omnipredictors Exist (Gopalan et al., 2022))**.** Let $\mathcal{L}_{\mathrm{cvx}}$ be the set of all convex, 1-Lipschitz loss functions. For any data distribution $\mathcal{D}$, hypothesis class $\mathcal{H}$, and $\varepsilon \geq 0$, there exist efficient $(\mathcal{L}_{\mathrm{cvx}}, \mathcal{H}, \varepsilon)$-omnipredictors. Further, such omnipredictors can be learned using a weak agnostic learner for $\mathcal{H}$, from $m = O(d_{\mathcal{H}} \cdot \mathrm{poly}(1/\varepsilon))$ samples from $\mathcal{D}$, where $d_{\mathcal{H}}$ represents the VC (or fat-shattering) dimension of $\mathcal{H}$.

The original construction of efficient omnipredictors follows as a consequence of *multicalibration*, a powerful notion for learning introduced in the algorithmic fairness literature by Hébert-Johnson et al. (2018). Even though multicalibration is defined without any reference to loss minimization, Gopalan et al. (2022)

[1] For example, the optimal prediction for ℓ_2 is $\mathbf{Pr}[y = 1|x]$, whereas for ℓ_1, it's the most likely outcome in $\{0, 1\}$.

[2] Technically, we say an omnipredictor is efficient if it has a $\mathrm{poly}(1/\varepsilon)$-sized circuit using oracle gates for $h \in \mathcal{H}$.

showed that, in fact, multicalibration implicitly guarantees loss minimization over every convex loss. While powerful, multicalibration is quite expensive in terms of sample complexity. For example, the sample complexity of omniprediction via multicalibration scales as $O(1/\varepsilon^{10})$ in the error parameter.

Subsequent work has investigated whether omniprediction can be achieved more efficiently than multicalibration. The work of Gopalan et al. (2023) introduced a general recipe, dubbed *Loss OI*, for achieving omnipredictors (for more general classes of loss functions) via two weaker conditions, *calibration* (Dawid, 1985) and *multiaccuracy* (Kim et al., 2019). Despite qualitative progress, the quantitative bounds for learning omnipredictors have not improved considerably. The state-of-the-art learning algorithms for efficient omnipredictors (even for subclasses of $\mathcal{L}_{\text{cvx}}$) still use sample complexity that scales as $O(1/\varepsilon^{10})$, considerably larger than the optimal ε-dependence for loss minimization of $\Theta(1/\varepsilon^2)$.

One concrete barrier to improved bounds for omniprediction is the reliance on calibration. At least in the online setting, calibration is known to require asymptotically more samples than loss minimization (Qiao and Valiant, 2021; Dagan et al., 2024). This "calibration bottleneck" has inspired researchers to look for alternative variants of calibration that are strong enough to give omniprediction-style guarantees, but weak enough to be achieved efficiently. Towards this end, Kleinberg et al. (2023) recently gave an algorithm that achieves simultaneous $O(\sqrt{T})$-regret for all *proper* losses; subsequently, Hu and Wu (2024) and Roth and Shi (2024) showed that similar bounds can be achieved for swap regret. These settings are non-contextual, so they do not directly guarantee omniprediction. Still, the results raise the prospect of simultaneous loss minimization at the price of minimizing a single loss. Indeed, Garg et al. (2024) showed a restricted setting in which online omniprediction is possible, albeit inefficiently, with near-optimal $\tilde{O}(\sqrt{T \log |H|})$ regret.

Despite a flurry of progress since its introduction, key questions about omniprediction remained unanswered. Most importantly, *what is the sample complexity of learning efficient omnipredictors?* In the online setting, what is the best omniprediction regret attainable? And how does the computational complexity of learning omnipredictors depend on the number of samples used? Underlying all of these questions is a methodological question about the relationship between calibration and omniprediction: *Is calibration necessary for efficient omniprediction, and if not, what weaker properties suffice?*

Our Contributions. In this work, we establish the near-optimal complexity of learning efficient omnipredictors. Remarkably, we show that for an extremely broad class of loss functions, there is essentially no cost to omniprediction compared to loss minimization. In both the online and distributional settings, we give oracle-efficient learning algorithms that establish upper bounds on the omniprediction regret (sample complexity, respectively) that match the lower bounds for minimization of a single loss function, up to low order factors. Our main contributions can be summarized as follows:

- **Proper Calibration.** We define a new variant of calibration—*proper calibration*—that is weaker than full calibration, but suffices for omniprediction. Following the Loss OI framework of Gopalan et al. (2023), we show that proper calibration actually characterizes "Decision OI" and, in doing so, show that proper calibration, paired with $\Delta\mathcal{L} \circ \mathcal{H}$-multiaccuracy, implies $(\mathcal{L}, \mathcal{H})$-omniprediction.

- **Near-Optimal Online Omniprediction.** We devise an online learning algorithm, based on Blackwell Approachability (Blackwell, 1956; Abernethy et al., 2011), that achieves proper calibration and multiaccuracy with near-optimal regret. Consequently, we obtain $\tilde{O}\left(\sqrt{T \cdot \log |\mathcal{H}|}\right)$ omniprediction regret, for any finite hypothesis class $\mathcal{H}$ and any class $\mathcal{L} \subseteq \mathcal{L}_{\text{BV}}$ of bounded variation losses; in

particular, $\mathcal{L}_{\mathrm{BV}}$ includes all Lipschitz losses $\mathcal{L}_{\mathrm{Lip}}$, their restriction to convex losses $\mathcal{L}_{\mathrm{cvx}}$, as well as all proper losses $\mathcal{L}_{\mathrm{prop}}$ and all bounded convex losses, irrespective of Lipschitz continuity.

- **Oracle-Efficiency.** We extend this basic approach, showing that, for any hypothesis class $\mathcal{H}$ (including infinite classes), online omniprediction reduces to *Online Weak Agnostic Learning*. This reduction establishes an upper bound on the omniprediction regret in terms of the sequential Rademacher complexity of the derived class $\Delta\mathcal{L} \circ \mathcal{H}$.

- **Efficient Offline Omnipredictors.** To achieve optimal omniprediction in the distributional setting, we develop an offline learning algorithm inspired by our online learning algorithm. In some sense, our algorithm can be viewed as a online-to-offline conversion, but running the conversion naively results in *inefficient* omniprediction. Instead, we devise a more sophisticated framework that maintains efficiency in samples, computation, and the resulting omnipredictor complexity simultaneously. In all, our algorithm returns an efficient randomized omnipredictor (that is evaluated by sampling one of $\mathrm{poly}(1/\varepsilon)$ hypotheses from $\mathcal{H}$ and postprocessing its output), using an *offline* ERM oracle, with sample complexity scaling near-optimally in the *offline* Rademacher complexity.

Organization. The remainder of the manuscript is organized as follows. The bulk of the introduction is dedicated to a detailed overview of our contributions (Section 1.1). We highlight our main results, emphasizing an intuitive understanding of our techniques. We conclude the introduction with a discussion of related works and open directions (Section 1.2). Beginning in Section 2, we give a complete presentation of our results, including formal definitions and proofs. Throughout Section 1.1, we aim to give pointers to the formal presentations of results in the body of the manuscript.

1.1 Overview of Contributions

Basic Preliminaries. We work in both an online and distributional prediction setting: inputs $x \in \mathcal{X}$ come from a discrete domain and outcomes $y \in \mathcal{Y} = \{0, 1\}$ are binary. Omniprediction aims to learn a single predictor $p : \mathcal{X} \to [0, 1]$ that guarantees loss minimization for every loss within a class $\mathcal{L} \subseteq \{\ell : [0, 1] \times \mathcal{Y} \to \mathbb{R}\}$ compared to the best hypothesis from a class $\mathcal{H} \subseteq \{h : \mathcal{X} \to [0, 1]\}$. In the distributional setting, we assume a fixed but unknown distribution $\mathcal{D}$ supported on $\mathcal{X} \times \mathcal{Y}$; when not specified, expectations are taken over $\mathcal{D}$.

We denote the set of all bounded loss functions[3] as $\mathcal{L}_{\mathrm{all}} = \{\ell : [0, 1] \times \mathcal{Y} \to [-1, 1]\}$ and consider loss classes defined based on functional properties (in the first argument). These loss classes include bounded variation $\mathcal{L}_{\mathrm{BV}}$, 1-Lipschitz $\mathcal{L}_{\mathrm{lip}}$, and (Lipschitz) convex losses $\mathcal{L}_{\mathrm{cvx}}$, with the following inclusions.

$$\mathcal{L}_{\mathrm{all}} \supsetneq \mathcal{L}_{\mathrm{BV}} \supsetneq \mathcal{L}_{\mathrm{lip}} \supsetneq \mathcal{L}_{\mathrm{cvx}}$$

One very important class of losses are the proper losses $\mathcal{L}_{\mathrm{proper}}$. A loss function ℓ is *proper* if for $y \sim \mathrm{Ber}(v)$, predicting v is an optimal strategy; that is, $\mathbf{E}[\ell(v, y)] \leq \mathbf{E}[\ell(u, y)]$ for any $u \in [0, 1]$. Note that the class of bounded proper losses $\mathcal{L}_{\mathrm{proper}}$ is a subset of $\mathcal{L}_{\mathrm{BV}}$, but incomparable to $\mathcal{L}_{\mathrm{lip}}$.

A key object in the study of omniprediction is the *discrete derivative* of a loss,

$$\Delta\ell(v) = \ell(v, 1) - \ell(v, 0)$$

[3]Throughout, we restrict our attention to measurable loss functions.

which intuitively captures whether ℓ distinguishes between $y = 1$ and $y = 0$ at a given prediction $v \in [0, 1]$.

Calibration is an essential notion for our discussion. We work with the generic notion of *weighted calibration*, which allows us to instantiate different variants easily.

Definition (Weighted Calibration Error)**.** Fix a class of functions $\mathcal{W} \subseteq \{w : [0, 1] \to [-1, 1]\}$, called *weight functions*. The $\mathcal{W}$-weighted calibration error is defined distributionally for any predictor $p : \mathcal{X} \to [0, 1]$, and sequentially, for any sequence of predictions, contexts, and outcomes $\mathbf{p}, \mathbf{x}, \mathbf{y}$, as follows.

$$\mathcal{W}\text{-CalErr}(p) = \sup_{w \in \mathcal{W}} \left| \mathbf{E}[w(p(x)) \cdot (y - p(x))] \right| \qquad \mathcal{W}\text{-CalErr}(\mathbf{p}, \mathbf{x}, \mathbf{y}) = \sup_{w \in \mathcal{W}} \left| \sum_{t=1}^{T} w(p_t(x_t))(y_t - p_t(x_t)) \right|$$

The standard notion of calibration (ℓ_1-calibration) corresponds to taking $\mathcal{W}_{\text{all}} = \{w : [0, 1] \to [-1, 1]\}$. Threshold functions are a key weight class, where $\mathcal{W}_{\text{Th}} = \{\text{Th}_\theta : \theta \in [0, 1]\}$ for $\text{Th}_\theta(p) = \text{sgn}(\theta - p)$.

Another important notion in the development of omnipredictors is multiaccuracy (Hébert-Johnson et al., 2018; Kim et al., 2019). Multiaccuracy is parameterized by a hypothesis class $\mathcal{H}$ and guarantees that the residual in predictions have no nontrivial correlation with any $h \in \mathcal{H}$.

Definition (Multiaccuracy)**.** Fix a hypothesis class $\mathcal{H} \subseteq \{h : \mathcal{X} \to [-1, 1]\}$. The $\mathcal{H}$-multiaccuracy error is defined distributionally for any predictor $p : \mathcal{X} \to [0, 1]$, and sequentially, for any sequence of predictions, contexts, and outcomes $\mathbf{p}, \mathbf{x}, \mathbf{y}$, as follows.

$$\mathcal{H}\text{-MAErr}(p) = \sup_{h \in \mathcal{H}} \left| \mathbf{E}[h(x) \cdot (y - p(x))] \right| \qquad \mathcal{H}\text{-MAErr}(\mathbf{p}, \mathbf{x}, \mathbf{y}) = \sup_{h \in \mathcal{H}} \left| \sum_{t=1}^{T} h(x_t)(y_t - p_t(x_t)) \right|$$

A complete set of preliminaries is given in Section 2.

1.1.1 Decision OI as Proper Calibration

Our approach to omniprediction follows the framework put forth by Gopalan et al. (2023) of Loss Outcome Indistinguishability (OI). As our first contribution, we give a novel characterization of one of the key components of the framework, Decision OI, in terms of a notion, which we call *proper calibration*. We develop this notion and its properties in Section 3. In particular, working with proper calibration reveals a more efficient scheme for achieving omniprediction, as we overview next.

The OI paradigm (Dwork et al., 2021) frames learning as indistinguishability. A predictor $p : \mathcal{X} \to [0, 1]$ satisfies OI if outcomes generated based on p "look like" real-world outcomes. Concretely, OI compares samples from the real world $(x, y) \sim \mathcal{D}$ and modeled samples $(x, \tilde{y})$ where $\tilde{y} \sim \text{Ber}(p(x))$ is resampled based on the predictor p. Loss OI guarantees $(\mathcal{L}, \mathcal{H})$-omniprediction via two sub-conditions, Hypothesis OI and Decision OI, using efficient tests defined by the losses $\ell \in \mathcal{L}$ and hypotheses $h \in \mathcal{H}$.

$$\textbf{Hypothesis OI:} \quad \mathop{\mathbf{E}}_{x,y\sim\mathcal{D}}[\ell(h(x), y)] \approx_\varepsilon \mathop{\mathbf{E}}_{\substack{x\sim\mathcal{D} \\ \tilde{y}\sim\text{Ber}(p(x))}}[\ell(h(x), \tilde{y})] \qquad \forall \ell \in \mathcal{L}, h \in \mathcal{H} \tag{1}$$

$$\textbf{Decision OI:} \quad \mathop{\mathbf{E}}_{x,y\sim\mathcal{D}}[\ell(k_\ell \circ p(x), y)] \approx_\varepsilon \mathop{\mathbf{E}}_{\substack{x\sim\mathcal{D} \\ \tilde{y}\sim\text{Ber}(p(x))}}[\ell(k_\ell \circ p(x), \tilde{y})] \qquad \forall \ell \in \mathcal{L} \tag{2}$$

Under these OI conditions, omniprediction follows immediately.

Theorem (Gopalan et al. (2023))**.** If p satisfies $(\mathcal{L}, \mathcal{H}, \varepsilon)$-Hypothesis OI and $(\mathcal{L}, \varepsilon)$-Decision OI, then p is an $(\mathcal{L}, \mathcal{H}, 2\varepsilon)$-omnipredictor.

This OI argument follows by switching from expectations in the real world (i.e., the LHS of (1) and (2)) to the modeled world (RHS). In the modeled world, outcomes are sampled from our predictor $\tilde{y} \sim \text{Ber}(p(x))$, so $k_\ell \circ p(x)$ is the *statistically-optimal* predictor; thus, in the real world, $\mathbf{E}[\ell(h(x), y)] \geq \mathbf{E}[\ell(k_\ell \circ p(x), y)] - 2\varepsilon$ by OI (losing an additive ε to switch from the real world to the modeled world and back).

Gopalan et al. (2023) go on to show that Hypothesis OI is equivalent to a certain multiaccuracy condition, for the class $\Delta\mathcal{L} \circ \mathcal{H} = \{\Delta\ell \circ h : \ell \in \mathcal{L},\ h \in \mathcal{H}\}$. They also show how the Decision OI error can be expressed as the following weighted calibration condition.

$$\Big| \mathop{\mathbf{E}}_{x,y\sim\mathcal{D}}[\ell(k_\ell \circ p(x), y)] - \mathop{\mathbf{E}}_{\substack{x\sim\mathcal{D} \\ \tilde{y}\sim\text{Ber}(p(x))}}[\ell(k_\ell \circ p(x), \tilde{y})] \Big| = \Big| \mathop{\mathbf{E}}_{x,y\sim\mathcal{D}}[\Delta\ell(k_\ell \circ p(x)) \cdot (y - p(x))] \Big| \tag{3}$$

While standard calibration implies this weighted calibration condition, we argue that it can be simplified into the following notion, based on proper losses.

Definition (Proper Calibration)**.** Let $\mathcal{W}_{\text{proper}} = \{\Delta\ell : \ell \in \mathcal{L}_{\text{proper}}\}$. A predictor $p : \mathcal{X} \to [0, 1]$ is ε-proper calibrated if $\mathcal{W}_{\text{proper}}$-CalErr$(p) \leq \varepsilon$; that is,

$$\sup_{\ell\in\mathcal{L}_{\text{proper}}} \Big| \mathbf{E}[\Delta\ell(p(x)) \cdot (y - p(x))] \Big| \leq \varepsilon$$

At first, this notion may seem insufficient to deal with improper losses. But when we compose a non-proper loss $\ell \in \mathcal{L}$ with the optimal post-processing k_ℓ, we can effectively treat the loss as proper. Formally, for any loss ℓ, there exists a proper loss $\ell_{\text{proper}} \in \mathcal{L}_{\text{proper}}$ such that $\Delta\ell(k_\ell(\cdot)) = \Delta\ell_{\text{proper}}(\cdot)$. The restriction to proper calibration allows us to exploit structural properties of proper losses. In particular, as in (Li et al., 2022; Kleinberg et al., 2023), we lean on a characterization of proper losses in terms of threshold functions. In all, we can show the following characterization of Decision OI in terms of proper calibration, and in terms of threshold-weighted calibration, which can be achieved efficiently.

Theorem 1. $\mathcal{L}_{\text{all}}$-Decision OI, Proper Calibration, and $\mathcal{W}_{\text{Th}}$-calibration are equivalent. Formally, for any predictor $p : \mathcal{X} \to [0, 1]$ (or sequence of predictions $\mathbf{p}$), the errors can be related as follows:

$$\mathcal{W}_{\text{Th}}\text{-CalErr}(p) \leq \mathcal{W}_{\text{proper}}\text{-CalErr}(p) = \mathcal{L}_{\text{all}}\text{-DecOIErr}(p) \leq 2 \cdot \mathcal{W}_{\text{Th}}\text{-CalErr}(p)$$

Thus, if p is $(\Delta\mathcal{L} \circ \mathcal{H}, \varepsilon)$-multiaccurate and $(\mathcal{W}_{\text{Th}}, \varepsilon)$-calibrated, then p is an $(\mathcal{L}, \mathcal{H}, 3\varepsilon)$-omnipredictor. Further, there exists an online algorithm that guarantees $O(\sqrt{T \log T})$ proper calibration regret.

With this characterization in hand, we can enforce proper calibration (and thus Decision OI) by auditing the predictor with threshold functions. Incorporating proper calibration—rather than ℓ_1-calibration—into the algorithmic framework of Gopalan et al. (2023) results in statistical improvements, but does not realize sample optimality. For completeness, we analyze this approach in Appendix C. Importantly, as we describe in the next section, we can enforce proper calibration simultaneously with multiaccuracy in the online setting to obtain near-optimal omniprediction regret, which we subsequently leverage to establish statistical near-optimality in the offline setting.

1.1.2 Blackwell Approachability for Online Omniprediction

In Section 4, we describe our strategy to learn predictions that obtain near-optimal omniprediction regret. Our algorithmic approach to online omniprediction is based on Blackwell Approachability (Blackwell, 1956) to solve a vector-valued game, defined by the proper calibration and multiaccuracy constraints. At a high level, we employ a framework developed by Abernethy et al. (2011) to use multiplicative weights over the set of constraints defined by the vector-valued game. Within this framework, we devise an explicit "halfspace oracle" used by the forecaster to play optimally given the dual weights. We begin with an overview of the intuition behind our approach, and describe our oracle-efficient implementation in the subsequent section.

To build intuition, we focus on the omniprediction setting where the loss class $\mathcal{L}$ and hypothesis class $\mathcal{H}$ are finite. As discussed above, our learning goal is to achieve low regret for proper calibration and online multiaccuracy simultaneously.[4] Per Theorem 1, the multiaccuracy and proper calibration regret necessary for omniprediction can be expressed as follows.

$$\textbf{Multiaccuracy:} \quad \max_{\substack{\ell\in\mathcal{L},\\ h\in\mathcal{H}}} \left|\sum_{t=1}^{T} \Delta\ell \circ h(x_t)(y_t - p_t)\right| \qquad \textbf{Proper Calibration:} \quad \sup_{\theta\in[0,1]} \left|\sum_{t=1}^{T} \mathrm{Th}_\theta(p_t)(y_t - p_t)\right|$$

In other words, the learner's job is to choose a sequence of predictions $\mathbf{p}$ that, when playing against an adversarial sequence of contexts $\mathbf{x}, \mathbf{y}$, guarantee low regret over a *worst-case* choice over loss-hypothesis pairs $(\ell, h) \in \mathcal{L} \times \mathcal{H}$ (multiaccuracy) and thresholds $\theta \in [0, 1]$ (proper calibration). In such a setting, we can formulate the learner's task as a Blackwell Approachability game with vector-valued payoffs.

Concretely, we can imagine the following finite-dimensional payoff vector $\vec{u}$. Given a prediction p, input $x \in \mathcal{X}$, and outcome $y \in \{0, 1\}$, the multiaccuracy constraints are indexed by loss-hypothesis pairs for each $\ell \in \mathcal{L}$ and $h \in \mathcal{H}$ and signs $s \in \{+, -\}$, and the proper calibration constraints are indexed by (appropriately-discretized) thresholds $\theta \in \{0, \varepsilon, \ldots, 1\}$, also signed by $s \in \{+, -\}$:

$$\textbf{MA:}\ \vec{u}_{\ell,h,s}(p, x, y) = s \cdot \Delta\ell \circ h(x) \cdot (y - p) \qquad \textbf{PC:}\ \vec{u}_{\theta,s}(p, x, y) = s \cdot \mathrm{Th}_\theta(p) \cdot (y - p)$$

The learner's goal in playing this game is to make the payoff vector $\vec{u}$ "approach" the origin, driving the worst-case violation of any constraint towards 0. To achieve this approachability, we can run multiplicative weights over the coordinates in $\vec{u}$, to maintain a dual halfspace $\vec{w}$ to witness violations of the constraints; then, given the halfspace, the algorithm computes an explicit optimal strategy to hedge against the choice of outcome $y_t \in \{0, 1\}$. Concretely, given a context x_t, we consider the following weighted function f, which (as a function of p), maps predictions on the interval $[0, 1]$ to the range $[-1, 1]$.

$$f(x_t, p) = \sum_{\ell,h,s\in\{\pm\}} w_{\ell,h,s} \cdot s \cdot \Delta\ell \circ h(x_t) + \sum_{\theta,s\in\{\pm\}} w^t_{\theta,s} \cdot s \cdot \mathrm{Th}_\theta(p) \tag{4}$$

Intuitively, $f(x_t, p) \cdot (y - p)$ captures the error (in multiaccuracy or proper calibration) that the learner may incur from predicting p on outcome y. Note that, for a fixed p, the adversary may choose the sign of this error through the choice of y. Thus, to minimize the potential error incurred—regardless of the outcome y—our algorithm plays a mixture between adjacent predictions p and p' where $f(x_t, p) \le 0$ and $f(x_t, p') > 0$, so that potential negative and positive error cancel in expectation.[5] By choosing the prediction interval appropriately, we can guarantee that the combined regret grows slowly.

[4] While natural, we don't know of any prior work that has studied multiaccuracy in the online setting. For completeness, we define the problem, and show how to achieve oracle-efficient optimal regret bounds in Appendix B.2.

[5] Note that the constraint functions defined by Th_θ are not continuous, so $f(x_t, \cdot)$ need not have a zero between p and p'.

Theorem 2. There exists an online algorithm that for any finite class of bounded loss functions $\mathcal{L}$ and finite hypothesis class $\mathcal{H}$, guarantees expected $(\mathcal{L}, \mathcal{H})$-omniprediction regret $O\left(\sqrt{T \log(|\mathcal{H}|\,|L|\,T)}\right)$.

1.1.3 Oracle-Efficient Online Omniprediction

Of course, the framework described above suffers from tracking weights for each multiaccuracy constraint explicitly. To achieve omniprediction more efficiently, or for infinite loss/hypothesis classes, we need a more sophisticated approach. To achieve these goals, we must generalize the algorithmic approach to avoid explicit dependence on a finite class of loss functions $\mathcal{L}$ and hypotheses $\mathcal{H}$. We will do this using an *Online Weak Agnostic Learner* for the class $\Delta\mathcal{L} \circ \mathcal{H}$, introduced in (Chen et al., 2012; Brukhim et al., 2020; Beygelzimer et al., 2015).

Our online algorithm for achieving $(\mathcal{L}, \mathcal{H})$-omniprediction consists of an interaction between two sub-algorithms.

- The first algorithm is an Online Weak Agnostic Learner (OWAL). The OWAL is responsible for producing a sequence of adaptively-chosen functions $q_1, \ldots, q_T : \mathcal{X} \to [-1, 1]$ such that enforcing a sequential multiaccuracy condition with respect to the q_t implies low multiaccuracy regret with respect to all of $\Delta\mathcal{L} \circ \mathcal{H}$. This online sparsification task is reminiscent of "scaffolding sets" problem, studied by Burhanpurkar et al. (2021) in the offline setting.
- The second algorithm is an Augmented Proper Calibration (APCAL). The APCAL produces a sequence of prediction functions p_t with low proper calibration regret on the sequence of (x_t, y_t); simultaneously, it enforces multiaccuracy with respect to the sequence of tests provided by the OWAL.

In more detail, these sub-algorithms satisfy the following semantics.

Online Weak Agnostic Learner. We use the OWAL abstraction to produce a sequence of functions such that $\{q_t\}$-multiaccuracy implies online $\Delta\mathcal{L} \circ \mathcal{H}$-multiaccuracy. Specifically, the OWAL solves the following online learning task: identify a sequence of functions $q_1, \ldots, q_T : \mathcal{X} \to [-1, 1]$ whose sequential multiaccuracy with respect to the sequence p_t is at least as large as the multiaccuracy violation of every $c \in \Delta\mathcal{L} \circ \mathcal{H}$ in hindsight:

$$\max_{c \in \Delta\mathcal{L} \circ \mathcal{H}} \sum_{t=1}^{T} c(x_t)(y_t - p_t(x_t)) \leq \sum_{t=1}^{T} q_t(x_t)(y_t - p_t(x_t)) + \text{OracleReg}_{\Delta\mathcal{L} \circ \mathcal{H}} \tag{5}$$

In other words, auditing against the sequence of T different q_t test functions (one at each time step) guarantees that the overall multiaccuracy regret is bounded.

Augmented Proper Calibration. The APCAL algorithm takes in a sequence of data $(x_1, y_1), \ldots, (x_T, y_T)$, and is responsible for producing a sequence of predictor functions $p_1, \ldots, p_T : \mathcal{X} \to [0, 1]$ that satisfy proper calibration over the given sequence. Additionally, the proper calibrator's input is augmented to receive a sequence of functions $q_1, \ldots, q_T : \mathcal{X} \to [-1, 1]$, and is responsible for simultaneously ensuring a sequential multiaccuracy guarantee with respect to these functions. In all, the APCAL is required to

guarantee the following regret bounds on its sequence.

$$\sup_{\theta\in[0,1]}\left|\sum_{t=1}^{T}\mathrm{Th}_\theta(p_t(x_t))(y_t-p_t(x_t))\right|\le\tilde{O}\left(\sqrt{T}\right)\qquad\text{and}\qquad\sum_{t=1}^{T}q_t(x_t)(y_t-p_t(x_t))\le\tilde{O}\left(\sqrt{T}\right)\tag{6}$$

Our implementation of the APCAL algorithm follows a similar approach to the finite omniprediction algorithm described earlier. As before, the algorithm runs multiplicative weights over each proper calibration constraint, indexed by the thresholds $\theta \in \{0, \varepsilon, \ldots, 1\}$. In contrast, however, rather than maintaining a weight per multiaccuracy constraint, our APCAL maintains a *single* weight for multiaccuracy. This weight determines the importance (and is updated according to violations) of the sequence of q_t functions that it receives from the OWAL.

Our eventual algorithm is still based on Blackwell Approachability, but approaches multiaccuracy implicitly through the sparse set of constraints provided by the OWAL. Specifically, in place of the high-dimensional payoff vector $\vec{u}(p, x, y)$ that enforced multiaccuracy constraints in the algorithm sketched above, we substitute the scalar $q_t(x)(y - p)$ where q_t is the output of the OWAL at time t. (The proper calibration constraints are still enforced using a $2T$-dimensional vector $\vec{u}_{\theta,s}(p, x, y)$ as before.) Then, at each step, our online algorithm bases its decision on a new function $f(x_t, p)$.

$$f(x_t,p)=w_{\mathrm{ma}}\cdot q_t(x_t)+\sum_{\theta,s\in\{\pm\}}w^t_{\theta,s}\cdot s\cdot\mathrm{Th}_\theta(p)\tag{7}$$

Instead of maintaining weights over $|\mathcal{H}| \times |\mathcal{L}| + 2T$ multiaccuracy and proper calibration constraints to compute the function as defined in Equation (4), we have collapsed all of the multiaccuracy constraints into a single dimension represented by the weight w_{ma}, thereby reducing the number of weights the algorithm must maintain to $2T + 1$. To the best of our knowledge, this method of "sparsifying" Blackwell Approachability by outsourcing many dimensions of the payoff vector to an auxiliary algorithm (in this case, the OWAL) that detects constraint violations in those dimensions is novel and may be of independent interest.

Once we implement each of these components, the online omniprediction guarantee follows immediately. Combining the guarantees from the APCAL and OWAL, we obtain an oracle-efficient online learning algorithm with the following properties.

Theorem 3. There exists an oracle-efficient online algorithm that for any class of loss functions $\mathcal{L}$ and any hypothesis class $\mathcal{H}$, given an online weak agnostic learner for $\Delta\mathcal{L} \circ \mathcal{H}$, guarantees expected $(\mathcal{L}, \mathcal{H})$-omniprediction regret $O\left(\sqrt{T \log T} + \mathrm{OracleReg}_{\Delta\mathcal{L}\circ\mathcal{H}}(T)\right)$.

Basis Decompositions for Infinite Loss Classes. As in prior works, given an infinite class of loss functions $\mathcal{L}$, we obtain omniprediction by designing a basis $\mathcal{G}$ that allows us to get uniform approximations to $\Delta\ell \in \Delta\mathcal{L}$. Then, using this basis in place of $\Delta\mathcal{L}$, we obtain omniprediction via a weak learner for $\mathcal{G} \circ \mathcal{H}$.

Using and refining bases developed by Gopalan et al. (2023, 2024b), we give online omniprediction guarantees for notable classes of loss functions. We define and construct these bases for loss classes formally in Section 4, but give a high-level summary here. Concretely, we obtain omniprediction guarantees from OWAL as outlined in Table 1.

A few comments are in order. First, notationally, for two classes of functions, we use $\mathcal{G} \circ \mathcal{H}$ to refer to the class of compositions $\{g \circ h : g \in \mathcal{G},\ h \in \mathcal{H}\}$. The oracles we need are for hypothesis classes derived by

Loss Class	OWAL Oracle
GLM Losses $\mathcal{L}_{\text{GLM}}$	$\mathcal{H}$
1-Lipschitz, Convex $\mathcal{L}_{\text{cvx}}$	$\text{ReLU}^{1/T} \circ \mathcal{H}$
1-Lipschitz $\mathcal{L}_{\text{lip}}$	$\text{Th}^{1/T} \circ \mathcal{H}$
Proper $\mathcal{L}_{\text{proper}}$	$\text{Th} \circ \mathcal{H}$
Bounded Variation $\mathcal{L}_{\text{BV}}$	$\text{Th} \circ \mathcal{H}$

Table 1: Summary of Loss Classes $\mathcal{L}$ and the associated Online Weak Agnostic Learning Oracle sufficient to achieve $(\mathcal{L}, \mathcal{H})$-omniprediction for hypothesis class $\mathcal{H}$ via Algorithm 3.

composition with $\mathcal{H}$. Specifically, we consider applying ReLU's and threshold functions.

$$\text{ReLU} = \{\text{ReLU}_\theta(\cdot) : \theta \in [0,1]\} \text{ where } \text{ReLU}_\theta(p) = \max\{0, p - \theta\}$$
$$\text{Th} = \{\text{Th}_\theta(\cdot) : \theta \in [0,1]\} \text{ where } \text{Th}_\theta(p) = \text{sgn}(\theta - p).$$

Importantly, while these classes are defined for $\theta \in [0,1]$, we use the superscript notation to denote a fixed precision. For instance, Th^γ considers threshold functions Th_θ for $\theta \in \{0, \gamma, 2\gamma, \ldots, 1\}$.

We obtain omniprediction for Lipschitz, convex loss functions leveraging an OWAL for (fixed-precision) ReLU's over hypotheses in $\mathcal{H}$. Notably, this oracle is different (and indeed a stronger oracle) than that used by Gopalan et al. (2022) to obtain $(\mathcal{L}_{\text{cvx}}, \mathcal{H})$-omnipredictors; indeed, $\mathcal{H}$-multicalibration is sufficient for convex omniprediction and can be obtained using a weak agnostic learner for $\mathcal{H}$. But as discussed, this strategy is statistically less efficient. Per (Gopalan et al., 2023), we can achieve omniprediction for Generalized Linear Model losses using a learner for $\mathcal{H}$ (because $\Delta\mathcal{L}_{\text{GLM}} \circ \mathcal{H} = \mathcal{H}$).

The remaining loss classes considered use an oracle for some class of threshold functions over $\mathcal{H}$. Omniprediction for the weakest of these classes, Lipschitz losses, can be achieved using an online learning oracle for fixed-precision threshold functions. While bounded variation losses (and proper losses which are a subset $\mathcal{L}_{\text{proper}} \subseteq \mathcal{L}_{\text{BV}}$) also follow from an oracle for thresholds applied to $\mathcal{H}$, the basis necessary to obtain uniform approximations of $\ell \in \mathcal{L}_{\text{BV}}$, is actually uncountably infinite. Indeed, the basis consists of a V-shaped loss for *every* $v \in [0,1]$, not simply a discrete finite approximation of the interval. Thus, in principle, we require an oracle for learning arbitrarily precise thresholds over $h \in \mathcal{H}$.

1.1.4 Near-Optimal Offline Omnipredictors

Online-to-offline conversion is a classic tool from learning theory: given a data set sampled from the target distribution $\mathcal{D}$, simulate the online learner on the samples (in arbitrary order); then, output the uniform mixture over predictions suggested at each "timestep" of the online algorithm. With a near-optimal online algorithm for omniprediction regret, one may hope that a corresponding distributional learning algorithm would follow from a standard online-to-offline conversion. Running such a conversion naively, however, results in an *inefficient* omniprediction algorithm, in a number of ways.

- **Representation Complexity.** To achieve online omniprediction unconditionally (without assuming an oracle), our approach maintains explicit weights for each hypothesis $h \in \mathcal{H}$. Converting this online algorithm into an offline omnipredictor produces a circuit of enormous complexity, scaling linearly in $|\mathcal{H}|$. Ideally, our offline algorithm would prove the existence of efficient omnipredictors for all hypothesis classes.

- **Statistical Complexity.** The regret bounds we achieve are near-optimal in the online setting, but naturally, depend on the *sequential dimension* of the class $\Delta\mathcal{L} \circ \mathcal{H}$. The sequential dimension can be arbitrarily (even infinitely) larger than the statistical dimension (e.g., Littlestone vs. VC). In the standard conversion of the online learner, we inherit the dependence on the sequential dimension.
- **Computational Complexity.** Finally, in aiming for computational efficiency, our online algorithm relies on an online weak agnostic learner. To learn omnipredictors in the distributional setting, we may hope to reduce the task of learning offline omnipredictors to an empirical risk minimizer.

We address each of these inefficiencies. In particular, we develop two parallel strategies for adapting our online algorithm to the distributional setting. The first approach, which runs multiplicative weights over a finite cover of $\Delta\mathcal{L} \circ \mathcal{H}$, is computationally-inefficient, but achieves near-optimal statistical complexity *for every* class of loss functions $\mathcal{L}$. The second approach manages to achieve efficiency based on an (offline) Empirical Risk Minimizer (ERM), while achieving statistical optimality for the class of proper losses $\mathcal{L}_{\text{proper}}$ (or alternatively all bounded variation losses $\mathcal{L}_{\text{BV}}$). Technically, however, this result is incomparable to our first approach. Our generalization argument leverages properties of the class $\text{Th} \circ \mathcal{H}$, so we pay for the Rademacher complexity of $\text{Th} \circ \mathcal{H}$ even for loss classes that have a much simpler $\Delta\mathcal{L} \circ \mathcal{H}$ (e.g., convex losses $\mathcal{L}_{\text{cvx}}$).

Statistically-Efficient Offline Omnipredictors. To begin, in Section 6, we give a conversion strategy that addresses the representation complexity and statistical complexity. While online-to-batch conversions are standard in learning theory, bounding the resulting error in our setting turns out to be very subtle. As usual, the distributional error the offline omnipredictor suffers can be broken down in terms of the regret achieved by the online learner and the generalization of the empirical statistics to their distributional analogues. We discuss each contribution separately.

To give a sufficiently strong bound on the regret, we need to design an online learner for omniprediction that, when fed data from the distribution $\mathcal{D}$, achieves error that scales with the statistical complexity, rather than the sequential complexity of the derived class. For this goal, we draw from the literature on *Online Hybrid Learning*. In this framework, features are drawn i.i.d. from a distribution, while the labels may be adversarially chosen. Similarly, in our offline setting, although the features are i.i.d., the labels $y_t - p_t(x_t)$ are adaptively chosen since the $p_1, \ldots, p_T$ are constructed in an online manner. Wu et al. (2022) and Lazaric and Munos (2009) demonstrate that applying a multiplicative weights algorithm over a "stochastic cover" of the hypothesis class yields an online hybrid learner with near-optimal dependence on the offline dimension of the hypothesis class for any convex, Lipschitz loss. Since the loss in our OWAL is convex and 1-Lipschitz, the statistical result follows immediately. While this strategy requires an explicit (inefficient) execution of multiplicative weights over the "stochastic cover" of hypotheses, the representation complexity arises from the fact that at each timestep, a single hypothesis is sampled. In all, the online-to-offline conversion outputs a distribution supported on $T = \text{poly}(1/\varepsilon)$ (postprocessed) hypotheses.

With an algorithm that allows us to achieve low regret on the empirical statistics, we need to prove generalization. Recall that in the online setting, our strategy was to bound omniprediction regret above by proper calibration error plus multiaccuracy error. However, upper bounds on multiaccuracy error in the sequential setting don't generalize readily to the distributional setting, because the "labels" used to define multiaccuracy error are the residuals, $y_t - p_t(x_t)$, rather than the raw labels y_t themselves. Hence, even

when the training set $\{(x_t, y_t)\}_{t=1}^T$ consists of i.i.d. draws from $\mathcal{D}$, the residuals $y_t - p_t(x_t)$ depend on an entire initial segment of the training set because p_t is trained using an online algorithm that sees the examples $(x_1, y_1), \ldots, (x_{t-1}, y_{t-1})$. This statistical adaptivity makes it very difficult (if not impossible) to prove that the multiaccuracy error converges to its distributional quantity.

To circumvent this difficulty, rather than proving separate generalization bounds for proper calibration error and multiaccuracy error, we prove generalization bounds for the statistics that are actually essential for omniprediction. Namely, we give generalization for the two expected losses that appear in the definition of omnipredictors: the loss of the best hypothesis in $\mathcal{H}$ and the loss obtained by postprocessing the omnipredictor's predictions using k_ℓ. The first of these two losses has no dependence on the sequence of predictors selected by our algorithm, so its generalization bound follows by standard Rademacher complexity arguments. The second generalization bound requires more care. Here, we make use of the fact that loss of the post-processed predictions, $\ell(k_\ell(p(x)), y)$, is equivalent to a proper loss $\ell_{\text{proper}}(p(x), y)$, which in turn is a weighted combination of "V-shaped losses" as in (Li et al., 2022; Kleinberg et al., 2023). In fact, since our omnipredictor always outputs predictions in the discrete set $[1/T] = \{0, 1/T, 2/T, \ldots, 1\}$, we argue that we only need to prove generalization for each of the T+1 V-shaped losses indexed by this set. This set of losses is small enough that a simple martingale argument suffices to conclude the generalization bound.

Ultimately, after running our online-to-offline conversion, the algorithm returns a randomized predictor, constituting a distribution over poly$(1/\varepsilon)$ deterministic predictors each obtained by postprocessing the output of a hypothesis from $\mathcal{H}$. Throughout the manuscript, we abbreviate this property by writing that the predictor "mixes over poly$(1/\varepsilon)$ postprocessed hypotheses from $\mathcal{H}$." In all, we derive the following theorem, which holds for any $\mathcal{H}$ unconditionally.

Theorem 4. There exists an algorithm $\mathcal{A}$ that for any distribution $\mathcal{D}$ supported on $\mathcal{X} \times \{0, 1\}$, for any class of loss functions $\mathcal{L} \subseteq \mathcal{L}_{\text{BV}}$, any hypothesis class $\mathcal{H}$, and $\varepsilon > 0$, learns an $(\mathcal{L}, \mathcal{H}, \varepsilon)$-omnipredictor with the following properties:

- $\mathcal{A}$ returns a randomized omnipredictor that mixes over poly$(1/\varepsilon)$ postprocessed hypotheses from $\mathcal{H}$.
- $\mathcal{A}$ uses $m \leq \tilde{O}(d_{\Delta\mathcal{L}\circ\mathcal{H}}/\varepsilon^2)$ samples drawn i.i.d. from $\mathcal{D}$, where $d_{\Delta\mathcal{L}\circ\mathcal{H}}$ denotes the VC dimension of $\Delta\mathcal{L} \circ \mathcal{H}$ or the fat-shattering dimension at scale ε in the case of a real-valued class.

As in the online case, we can instantiate the result using an appropriate basis in place of $\Delta\mathcal{L}$. As such, for each of the loss-hypothesis class pairs highlighted in Table 1, the distributional algorithm depends on the corresponding offline statistical complexity of the class.

In other words, even though the result is computationally-inefficient, it establishes a near-optimal statistical complexity that adapts to the complexity of the loss class $\mathcal{L}$. If better bases are developed for $\Delta\mathcal{L}$, the statistical dependence of the result will adapt to depend on the complexity of the resulting class.

Oracle-Efficient Offline Omnipredictors. While the offline algorithm described above is sample-efficient and representation-efficient, it suffers in one respect: it is computationally-inefficient and runs in time linear in the number of hypotheses. Addressing this inefficiency, in Section 7, we devise an offline algorithm to learn efficient omnipredictors with respect to any loss class $\mathcal{L} \subseteq \mathcal{L}_{\text{BV}}$ and hypothesis class $\mathcal{H}$ that achieves near-optimal sample complexity and is computationally-efficient given an ERM oracle for the hypothesis class $\text{Th} \circ \mathcal{H}$. In all, we show the following theorem.

Theorem 5 (Oracle-Efficient Omniprediction)**.** There exists an oracle-efficient algorithm $\mathcal{A}$ that for any distribution $\mathcal{D}$ supported on $\mathcal{X} \times \{0,1\}$, for any class of loss functions $\mathcal{L} \subseteq \mathcal{L}_{\mathrm{BV}}$, any hypothesis class $\mathcal{H}$, and $\varepsilon > 0$, learns an $(\mathcal{L}, \mathcal{H}, \varepsilon)$-omnipredictor with the following properties:

- $\mathcal{A}$ returns a randomized omnipredictor that mixes over $\mathrm{poly}(1/\varepsilon)$ postprocessed hypotheses from $\mathcal{H}$.
- Given m samples, the error of the omnipredictor returned by $\mathcal{A}$ scales near-optimally with the *offline* Rademacher complexity $\varepsilon(m) = \tilde{O}\left(\mathsf{rad}_m(\mathrm{Th} \circ \mathcal{H})\right) + O\left(\sqrt{\ln m / m}\right)$.
- $\mathcal{A}$ is oracle-efficient, making $\mathrm{poly}(1/\varepsilon)$ calls to an *offline* ERM oracle for $\mathrm{Th} \circ \mathcal{H}$.

Our oracle-efficient offline algorithm is an adaptation of the online algorithm for achieving $(\mathcal{L}, \mathcal{H})$-omniprediction, which simultaneously maintains low proper calibration regret and low multiaccuracy regret. Recall that our online algorithm consists of an interaction between two sub-algorithms: the Augmented Proper Calibration (APCAL) and the Online Weak Agnostic Learner (OWAL). One way to achieve computational efficiency would be to make the OWAL computationally efficient with respect to an ERM oracle without losing sample efficiency. However, the problem of designing hybrid online learning algorithms (of which the OWAL is a special case) that are both oracle-efficient and sample-efficient has remained an open problem for many years. We bypass this challenge of implementing an oracle-efficient OWAL by introducing a different interface, a Distributional Online Weak Agnostic Learner (DOWAL), to replace the OWAL. The DOWAL is responsible for producing the sequence of functions such that $\{q_t\}$-multiaccuracy implies $\Delta\mathcal{L} \circ \mathcal{H}$-multiaccuracy over the distribution $\mathcal{D}$. More formally, the DOWAL must solve the following hybrid learning task: identify a sequence of functions $q_1, \ldots, q_T : \mathcal{X} \to [-1, 1]$ that witness the distributional multiaccuracy error (over $\mathcal{D}$) of the sequence p_t as effectively as the best fixed $c \in \Delta\mathcal{L} \circ \mathcal{H}$ in hindsight:

$$\max_{c \in \Delta\mathcal{L} \circ \mathcal{H}} \sum_{t=1}^{T} \mathop{\mathbf{E}}_{(x,y)\sim\mathcal{D}}[c(x)(y - p_t(x))] \leq \sum_{t=1}^{T} \mathop{\mathbf{E}}_{(x,y)\sim\mathcal{D}}[q_t(x)(y - p_t(x))] + \mathrm{Regret}_{\mathrm{dowal}}(T) \tag{8}$$

The key distinction between the OWAL and the DOWAL is that the OWAL provides a guarantee over the online sequence while the DOWAL provides a guarantee over the distribution.

Importantly, the oracle-efficient offline algorithm partitions its samples drawn from $\mathcal{D}$ into two sets of size $m = T$, denoted by D_{apcal} and D_{dowal}, and provides each sub-algorithm only with its half of the data. By splitting samples, and designing careful interfaces through which the sub-algorithms interact, we are able to maintain the martingale difference property we need for generalization of the proper calibration error, while using a uniform convergence bound to argue generalization for the multiaccuracy tests generated by the DOWAL. In particular, a martingale argument on the APCAL guarantee in Equation (6) leads to the following distributional guarantee

$$\sup_{\theta \in [0,1]} \left| \mathop{\mathbf{E}}_{(x,y)\sim\mathcal{D}}[\mathrm{Th}_\theta(p_t(x))(y - p_t(x))] \right| \leq \tilde{O}\left(\sqrt{T}\right) \qquad \text{and} \qquad \mathop{\mathbf{E}}_{(x,y)\sim\mathcal{D}}[q_t(x)(y - p_t(x))] \leq \tilde{O}\left(\sqrt{T}\right)$$

Combining this with Equation (8) allows us to conclude that the uniform mixture $\hat{\mathbf{p}}$ over the sequence of predictors $p_1, \ldots, p_T$ is approximately proper calibrated and $\Delta\mathcal{L} \circ \mathcal{H}$-multiaccurate over the distribution $\mathcal{D}$ with error that scales as $\tilde{O}\left(\sqrt{1/T}\right) + \frac{1}{T}\mathrm{Regret}_{\mathrm{dowal}}(T)$

Implementing the DOWAL via Low-Regret Learning The DOWAL must achieve the regret guarantee expressed in Equation (8). We implement the DOWAL by designing a procedure that satisfies a corresponding regret bound with respect to the *empirical* counterparts of the distributional expected values.

$$\max_{c \in \Delta\mathcal{L} \circ \mathcal{H}} \sum_{t=1}^{T} \mathop{\mathbf{E}}_{(x,y)\sim D_{\text{dowal}}} [c(x)(y - p_t(x))] \leq \sum_{t=1}^{T} \mathop{\mathbf{E}}_{(x,y)\sim D_{\text{dowal}}} [q_t(x)(y - p_t(x))] + \tilde{O}(\sqrt{T}) \tag{9}$$

Then, separately, we establish uniform convergence of the empirical quantities of interest to their distributional values.

The problem of selecting $q_1, \ldots, q_T$ to minimize the regret on the right side of (9) reduces to online linear optimization, using the fact that the distribution D_{dowal} has support size m. We embed each $c \in \Delta\mathcal{L} \circ \mathcal{H}$ as a vector in $[-1, 1]^m$ by evaluating $c(x)$ at each $x \in \{x_1, \ldots, x_m\}$, and we observe that in this embedding, two convenient properties hold.

1. The objective function $\mathbf{E}_{(x,y)\sim D_{\text{dowal}}}[c(x)(y - p_t(x))]$ is a linear function of the vector representing c.
2. An ERM oracle for $\Delta\mathcal{L}\circ\mathcal{H}$ enables us to optimize any linear function over the embedding of $\Delta\mathcal{L}\circ\mathcal{H}$.

Our implementation of the DOWAL solves the regret minimization problem embodied by (9) using a Follow-The-Regularized-Leader (FTRL) procedure. Although our online learning problem is m-dimensional, we show that its decision set and loss vectors have a structure that ensures a dimension-independent $O(\sqrt{T})$ regret bound for FTRL with the entropy regularizer. As for the oracle complexity of implementing FTRL, each iteration requires approximately minimizing a strongly convex function over the embedding of $\Delta\mathcal{L}\circ\mathcal{H}$ in $[-1, 1]^m$; we reduce this strongly convex minimization problem to a sequence of ERM oracle calls via the Frank-Wolfe procedure.

Uniform Convergence for the DOWAL. The DOWAL is initialized with a dataset D_{dowal} drawn from $\mathcal{D}$ and guarantees low empirical multiaccuracy regret. The key question is: how many samples must D_{dowal} contain in order to guarantee that the empirical quantities in Equation (9) converge to imply the distributional guarantee of Equation (8)?

To establish convergence, we analyze the Rademacher complexity of functions of the form $q_t(x)\cdot(y-p_t(x))$ that might arise in our computations. This analysis requires us to be very careful about the complexity of the functions that may be returned by each of the DOWAL (i.e., $q_1, \ldots, q_T$) and the APCAL (i.e., $p_1, \ldots, p_T$).

Specifically, by leveraging properties of the threshold basis for bounded variation losses, for any loss class $\mathcal{L} \subseteq \mathcal{L}_{\text{BV}}$, we are able to bound the generalization error for Equation (9) in terms of the Rademacher complexity of $\text{Th} \circ \mathcal{H}$ near-linearly. Our reliance on properties of thresholds over $\mathcal{H}$, however, means that we cannot improve the Rademacher dependence, even for simpler loss classes. Still, in all, we are able to show the uniform convergence guarantee for the DOWAL in terms of the Rademacher complexity of $\text{Th} \circ \mathcal{H}$, establishing the near-optimal sample complexity for $(\mathcal{L}, \mathcal{H})$-omniprediction with ERM oracle-efficiency.

1.2 Discussion of Results and Related Works

To conclude the introduction, we discuss the significance of our results and highlight some interesting questions that we leave open. We also contextualize our results within the prior work on omnipredictors.

The Complexity of Omnipredictors. In the distributional context, we learn omnipredictors that can be implemented with circuits of poly$(1/\varepsilon)$-size using oracle gates for $h \in \mathcal{H}$, qualitatively matching the circuit complexity of prior omnipredictor constructions (Gopalan et al., 2022, 2023). That said, there are key differences in the omnipredictors produced by our algorithm that achieves near-optimal statistical complexity and prior omnipredictors. In particular, our omnipredictors are actually *randomized*: our omnipredictors mix over a collection of efficient predictors. While the use of randomness can be removed in certain contexts, in generality, randomness is a critical aspect of the online-to-offline conversion. Naturally, this discrepancy suggests questions about the complexity of statistically-optimal omniprediction. *Can our omnipredictors be derandomized?* Or, *Is randomness necessary to achieve sample-optimal omnipredictors?*

Beyond sample and randomness complexity, there are a number of other measures on which we can compare constructions of omnipredictors. For instance, the boosting-based learning framework used in all prior omnipredictor constructions (Hébert-Johnson et al., 2018) produces *deep* omnipredictors, with poly$(1/\varepsilon)$ layers of computation. In contrast, our randomized omnipredictors are essentially as shallow as possible: we output a distribution supported on poly$(1/\varepsilon)$ predictors, each of which is implemented by a postprocessing of hypotheses from $\mathcal{H}$.

As the community understands the problem of omniprediction better, a complexity theory is emerging, where we can ask precise questions about what is and isn't possible along many axes of complexity measures. Precise accounting of complexity measures may be particularly interesting for applications of omnipredictors and multicalibrated predictors within complexity theory and pseudorandomness (Casacuberta et al., 2024). Our work makes significant strides in settling the complexity of omniprediction along important measures, but also raises a number of interesting questions for future research.

Towards Optimal Decision Making. This work adds to a growing collection of works investigating when strong omniprediction-style guarantees can be achieved with regret comparable to a single loss. The majority of the results in this area work in the online, non-contextual setting. A trend in these works is identifying alternatives to full sequential calibration, which inherently suffers $\Omega(T^{1/2+\varepsilon})$ regret, and until this year, was only known to be achievable in $O(T^{2/3})$ regret (Foster and Vohra, 1998; Dagan et al., 2024).

First in this line of work, Kleinberg et al. (2023) introduced the notion of U-Calibration, or simultaneous loss minimization with respect to all proper loss functions. Leveraging the V-shaped basis for proper losses (also studied by Li et al. (2022)) and a representation of the Hedge algorithm (Freund and Schapire, 1997) for agents with V-shaped losses as a composition of randomized prediction with postprocessing, they give a $O(\sqrt{T})$-regret online learner to achieve U-Calibration. Subsequent work established optimal U-Calibration regret bounds for the multi-class setting (Luo et al., 2024).

Hu and Wu (2024) study how to achieve swap regret bounds simultaneously for all proper losses (i.e., a strengthened version of U-Calibration to allow for a different choice of optimal action in hindsight per prediction interval). They introduce the notion of *calibrated decision loss* (CDL) and show how to achieve $O(\sqrt{T}\log T)$ regret for CDL, guaranteeing the same bounds for swap regret. In Appendix A, we show that CDL is actually incomparable to proper calibration. In this sense, CDL measures calibration error differently than proper calibration: CDL cannot be used to achieve omniprediction via Decision OI, and proper calibration does not give swap regret guarantees.

Most pertinent to our study, Garg et al. (2024) introduced the study of omniprediction in the online setting. They give oracle-efficient algorithms for a number of related problems, including the stronger problem of online multicalibration, which in turn gives the stronger guarantee of swap omniprediction for convex

losses (Gopalan et al., 2024a). The strength of their guarantees comes at a cost, obtaining $O(T^{3/4})$ swap regret. In addition to this oracle-efficient result, they also include a result that shows, in restricted settings of loss and hypothesis class, $\tilde{O}(\sqrt{T \log |\mathcal{H}|})$ omniprediction regret is information-theoretically possible.

Following Garg et al. (2024) and concurrent with the development of our work, Dwork et al. (2024) continued the study of online omniprediction. While their study is quite broad, motivated by fairness concerns in professional networks, Dwork et al. (2024) develop online kernel outcome indistinguishability as a core algorithmic primitive. They give an algorithm for online omniprediction, with respect to differentiable strongly convex losses (a subclass of $\mathcal{L}_{\text{cvx}}$) and hypotheses from a Reproducing Kernel Hilbert Space (RKHS), that attains $O(\sqrt{T})$ regret in its dependence on T. While they obtain an optimal dependence on T, the dependence on the RKHS is considerably inefficient; for instance, for the kernel of degree-d polynomials over $\mathcal{X} = \{0,1\}^n$, the regret scales as $O(\sqrt{T \cdot n^d})$, whereas our online algorithm would obtain $O(\sqrt{Td \log n})$ for an analogous omniprediction problem.

Finally, in closely related concurrent work, Hu et al. (2024) study the problem of omniprediction for Single Index Models, which generalize the goal of omniprediction for GLM losses. They achieve improved sample complexity of ε^{-4}, as a function of the approximation parameter. While their motivations and results are similar to ours, their techniques are very different, involving a new analysis of the Isotron algorithm (Kalai and Sastry, 2009; Kakade et al., 2011).

Omniprediction and Multi-Group Fairness. Since its introduction by Gopalan et al. (2022), omniprediction has been studied in numerous works. Gopalan et al. (2023) introduced Loss Outcome Indistinguishability as a method for obtaining omnipredictors, working within the Outcome Indistinguishability (OI) paradigm developed by Dwork et al. (2021, 2022). Within this work, the notion of Decision OI (which we establish is equivalent to Proper Calibration) draws directly from the work of Zhao et al. (2021), who introduced the related notion of Decision Calibration for the multi-class prediction setting. Omniprediction has also been studied in the context of constrained predictors (Hu et al., 2023; Globus-Harris et al., 2023a) and in other prediction settings, including regression (Gopalan et al., 2024b) and performative prediction (Kim and Perdomo, 2023).

The study of omniprediction is tightly connected to the study of multi-group fairness (Hébert-Johnson et al., 2018; Kim et al., 2019). While omniprediction has always been known to follow from multicalibration, recent work showed a connection in the reverse direction. Specifically, Gopalan et al. (2024a) study a stronger notion of omniprediction called *swap omniprediction*, where the omnipredictor p must compete against a benchmark where a different hypothesis $h \in \mathcal{H}$ may be chosen per loss function $\ell \in \mathcal{L}$ and level set of p. In other words, the constant prediction $p(x) = v$ must out-compete every $h \in \mathcal{H}$ with respect to every $\ell \in \mathcal{L}$ on the set of x such that $p(x) = v$. Gopalan et al. (2024a) demonstrate that Swap Omniprediction for convex losses actually characterizes Multicalibration. In fact, they show both notions are equivalent to Swap Squared Error Minimization, concurrently studied by Globus-Harris et al. (2023b).

1.3 Acknowledgements

We would like to enthusiastically thank Karthik Sridharan for many helpful discussions throughout the development of this work about online and statistical learning theory. In particular, Karthik provided invaluable resources and ideas in the development of our online-to-offline conversions.

2 Model and Preliminaries

Omniprediction for Binary Outcomes We are concerned with the binary prediction setting where there is a distribution $\mathcal{D}$ over pairs of feature vectors $x \in \mathcal{X}$ and binary outcomes $y \in \mathcal{Y} = \{0, 1\}$. With only sample access to this distribution, the goal is to learn a predictor $p : \mathcal{X} \to [0, 1]$ that outperforms a class of hypothesis functions $\mathcal{H} = \{h : \mathcal{X} \to [0, 1]\}$ (e.g decision trees, neural networks) over a range of loss functions $\mathcal{L} = \{\ell : [0, 1] \times \{0, 1\} \to [-1, 1]\}$. Formally, we define an omnipredictor as follows:

Definition 2.1 (Omnipredictor Gopalan et al. (2022))**.** Let $\mathcal{H}$ be a family of functions on $\mathcal{X}$ and let $\mathcal{L}$ be a family of loss functions. The predictor $p : \mathcal{X} \to [0, 1]$ is an $(\mathcal{L}, \mathcal{H}, \varepsilon)$-omnipredictor if for every $\ell \in \mathcal{L}$ there exists a function $k_\ell : [0, 1] \to [0, 1]$ so that

$$\mathop{\mathbf{E}}_{(x,y)\sim\mathcal{D}}[\ell(k_\ell(p(x)), y)] \leq \min_{h\in\mathcal{H}} \mathop{\mathbf{E}}_{(x,y)\sim\mathcal{D}}[\ell(k_\ell(h(x)), y)] + \varepsilon$$

One can think of an onmipredictor as a model of the world (or distribution) that is sufficient for the learner to perform at least as well as the best hypothesis in $\mathcal{H}$ with respect to the loss functions in $\mathcal{L}$. The post-processing function k_ℓ is defined as follows:

$$k_\ell(p) = \arg\min_{q\in[0,1]} \mathop{\mathbf{E}}_{y\sim\mathrm{Ber}(p)}[\ell(q, y)]$$

Online Omniprediction In the online version of the omniprediction, we will consider a sequential setting where each round $t \in [T]$, a context $x_t \in \mathcal{X}$ arrives. On observing x_t, a forecaster makes a prediction p_t of $\mathbf{E}[y_t|x_t]$ and then observes y_t which may be adversarily chosen. Equivalently, each timestep $t \in [T]$, the forecaster chooses a prediction function $p_t : \mathcal{X} \to [0, 1]$, and the adversary, unaware of the forecaster's choice, chooses a pair $(x_t, y_t) \in \mathcal{X} \times \mathcal{Y}$. We assume the adversary is fully aware of history H_t of predictions made by the forecaster up until timestep $t - 1$. We measure the performance of the forecaster over a range of loss functions in a class $\mathcal{L}$ and a benchmark hypothesis class $\mathcal{H}$. For a sequence of T predictions $\mathbf{p}$ and sequence of context, outcome pairs $\mathbf{x}, \mathbf{y}$, define the forecaster's regret as follows:

$$(\mathcal{L}, \mathcal{H})\text{-OmniRegret}(\mathbf{p}, \mathbf{x}, \mathbf{y}) = \mathbf{E}\left[\max_{\{h\in\mathcal{H},\ell\in\mathcal{L}\}} \sum_{t=1}^{T} \ell(k_\ell(p_t(x_t)), y_t) - \ell(h(x_t), y_t)\right]$$

where the expectation is over the randomness of the forecaster.

When the class of loss functions is the class of all bounded proper scoring losses, this objective becomes a contextual version of the U-calibration objective in Kleinberg et al. (2023).

Definition 2.2 (Multiaccuracy Error)**.** Let $C = \{c : \mathcal{X} \to [-1, 1]\}$ be a family of hypothesis functions. For a sequence of T predictions $\mathbf{p}$ and context, outcome pairs $\mathbf{x}, \mathbf{y}$, define

$$C\text{-MAErr}(\mathbf{p}, \mathbf{x}, \mathbf{y}) = \max_{c\in C} \left|\sum_{t=1}^{T} c(x_t)(y_t - p_t(x_t))\right|$$

We present the notion of weighted calibration error.

Definition 2.3 (Weighted Calibration Error)**.** Let $\mathcal{W} = \{w : [0,1] \to [-1,1]\}$ be a family of weight functions. For a sequence of T predictions $\mathbf{p}$ and context, outcome pairs $\mathbf{x}, \mathbf{y}$, define

$$\mathcal{W}\text{-CalErr}(\mathbf{p}, \mathbf{x}, \mathbf{y}) = \sup_{w \in \mathcal{W}} \left| \sum_{t=1}^{T} w(p_t(x_t))(y_t - p_t(x_t)) \right|$$

Note that when $\mathcal{W}$ is the set of all functions $w : [0,1] \to [-1,1]$, this becomes the ℓ_1-calibration error.

The key distinction between multiaccuracy error and weighted calibration error is that the functions in calibration error can also depend on the predictions of the predictor.

Loss Families, Hypothesis Classes and ERM Oracles The loss functions we consider are functions $\ell : [0,1] \times \{0,1\} \to \mathbb{R}$ that take a binary outcome and a prediction and assigns a real value $\ell(p, y)$. Let $\mathcal{L}_{\text{lip}}$ denote the set of losses ℓ that are 1-Lipschitz in p and $\mathcal{L}_{\text{cvx}}$ denote the set of losses convex in p.

Definition 2.4 (Proper Losses)**.** A loss function ℓ is said to be proper if

$$\mathop{\mathbf{E}}_{y \sim \text{Ber}(p^*)}[\ell(p^*, y)] \leq \mathop{\mathbf{E}}_{y \sim \text{Ber}(p^*)}[\ell(q, y)]$$

for all $q \in [0,1]$. Let $\mathcal{L}_{\text{proper}}$ denote the set of all such losses. Note that this is also the set of all losses for which $p \in k_\ell(p)$.

Definition 2.5 (Bounded Variation)**.** A function $f : [0,1] \to \mathbb{R}$ has *bounded variation* if the quantity

$$V(f) = \sup \left\{ \sum_{i=1}^{n} |f(x_i) - f(x_{i-1})| \,\middle|\, 0 = x_0 < x_1 < \cdots < x_n = 1 \right\}$$

is finite. The class $\mathcal{L}_{\text{BV}}$ of bounded variation losses consists of all loss functions $\ell(p, y)$ taking values in $[-1,1]$ that satisfy $V(\Delta\ell) \leq 2$.

Definition 2.6 (Discrete Derivative)**.**

$$\Delta\ell(t) = \ell(t,1) - \ell(t,0)$$

Observe that $\ell(t, y) = y\Delta\ell(t) + \ell(t,0)$. For a class of loss functions $\mathcal{L}$, we will refer to $\Delta\mathcal{L} = \{\Delta\ell : \ell \in \mathcal{L}\}$. We will characterize families of loss functions based on the complexity of the discrete derivate class.

Definition 2.7 (ERM Oracle)**.** Our algorithms will require a empirical risk minimization oracle for a specified class of hypothesis functions $\mathcal{H}$. This oracle will take as a input a sequence of samples $(x_1, y_1) \ldots, (x_T, y_T)$ and returns a hypothesis $h^* \in \mathcal{H}$ such that

$$h^* \in \arg\min_{h \in \mathcal{H}} \sum_{t=1}^{T} h(x_t) y_t$$

Online Learning and Blackwell's Approachability Theorem In online learning, the learner's objective is to select a sequence of actions from a given set such that the average regret asymptotically approaches zero, irrespective of the loss functions chosen by an adversary.

Blackwell's approachability theorem (Blackwell, 1956) generalizes repeated two-player zero-sum games to setting where payoffs are vector-valued. In this framework, at each time step t, Player 1 selects an action

$a_t \in A$, Player 2 responds with $b_t \in B$, and Player 1 receives a vector-valued payoff $u(a_t, b_t) \in \mathbb{R}^d$. The action sets A and B are compact convex subsets of finite-dimensional spaces, and the payoff function u is biaffine over $A \times B$. Player 1's goal is to ensure that the average payoff vector converges to a closed convex target set $S \subseteq \mathbb{R}^d$. Formally, given S, Player 1 chooses actions so that, regardless of Player 2's choices, the distance between the average payoff and S approaches zero as $T \to \infty$.

$$\texttt{dist}\left(\frac{1}{T}\sum_{t=1}^{T} u(a_t, b_t), S\right) \to 0 \quad \text{as} \quad T \to \infty \tag{10}$$

Player 1's actions can adapt based on previous outcomes. A set S is approachable if Player 1 can guarantee this convergence. Blackwell's theorem characterizes this by stating that S is approachable if and only if every closed halfspace containing S is approachable.

Many online learning problems can be effectively solved by a reduction to Blackwell's approachability theorem. In fact, Abernethy et al. (2011) show that approachability and no-regret learning are equivalent. In this paper, we reduce the online omniprediction problem to an approachability problem and apply the regret minimization reduction techniques from Abernethy et al. (2011) to solve the resulting approachability problem.

Combinatorial Dimensions for Learning Given a feature space $\mathcal{X}$ and a fixed distribution $D|_{\mathcal{X}}$, let $S = \{x_1, \ldots, x_m\}$ be a set of examples drawn i.i.d. from $D|_{\mathcal{X}}$. Furthermore, let $\mathcal{F}$ be a class of functions $f : \mathcal{X} \to [-1, 1]]$.

Definition 2.8 (α-shattering)**.** We say $\mathcal{F}$ α-shatters the set S if there exists values $v_1, \ldots, v_m \in [0, 1]$ such that for all $A \subseteq S$, there exists $f_A \in \mathcal{F}$ such that

$$\begin{aligned} \forall x_i \in A, \quad & f_A(x_i) \geq v_i + \alpha \\ \forall x_i \in S - A, \quad & f_A(x_i) \leq v_i - \alpha \end{aligned}$$

The fat shattering dimension of $\mathcal{F}$ at scale α is the size of the largest α-shattered set. For binary valued class $\mathcal{F} \subseteq \{0, 1\}^{\mathcal{X}}$, the VC dimension of $\mathcal{F}$ is the size of the largest $1/2$-shattered set.

Definition 2.9. The *empirical Rademacher complexity* of $\mathcal{F}$ is defined to be

$$\hat{\mathsf{rad}}_m(\mathcal{F}; S) = \mathbb{E}_\sigma\left[\sup_{f \in \mathcal{F}}\left(\frac{1}{m}\sum_{i=1}^{m} \sigma_i f(x_i)\right)\right]$$

where $\sigma_1, \ldots, \sigma_m$ are independent random variables uniformly chosen from $\{-1, 1\}$. We will refer to such random variables as Rademacher variables.

Definition 2.10. The *statistical Rademacher complexity* of $\mathcal{F}$ is defined as

$$\mathsf{rad}_m(\mathcal{F}) = \mathbb{E}_{S \sim \mathcal{D}}[\hat{\mathsf{rad}}_m(\mathcal{F}; S)]$$

The following is a well know result in learning theory:

Lemma 2.1. Fix distribution $D|_X$ and parameter $\delta \in (0,1)$. If $\mathcal{F} \subseteq \{f : \mathcal{X} \to [-1,1]\}$ and $S = \{x_1, \ldots, x_m\}$ is drawn i.i.d. from $D|_X$, then with probability $\geq 1-\delta$ over the draw of S, for every function $f \in \mathcal{F}$,

$$\mathbb{E}_D[f(x)] \leq \mathbb{E}_S[f(x)] + 2\mathsf{rad}_m(\mathcal{F}) + \sqrt{\frac{\ln(1/\delta)}{m}}. \tag{1}$$

In addition, with probability $\geq 1-\delta$, for every function $f \in \mathcal{F}$,

$$\mathbb{E}_D[f(x)] \leq \mathbb{E}_S[f(x)] + 2\hat{\mathsf{rad}}_m(\mathcal{F}) + 3\sqrt{\frac{\ln(2/\delta)}{m}}. \tag{2}$$

Lemma 2.2 (Sridharan (2010))**.** Fat-shattering dimension and Rademacher complexities are related as follows:

$$\tilde{\Omega}\left(\inf_{\alpha>0}\left\{4\alpha + \frac{12}{\sqrt{m}}\int_\alpha^1 \sqrt{K\,\mathrm{fat}_\delta(\mathcal{F})\log\frac{2}{\delta}}\,d\delta\right\}\right) \leq \mathsf{rad}_m(\mathcal{F}) \leq \inf_{\alpha>0}\left\{4\alpha + \frac{12}{\sqrt{m}}\int_\alpha^1 \sqrt{K\,\mathrm{fat}_\delta(\mathcal{F})\log\frac{2}{\delta}}\,d\delta\right\}$$

where $\tilde{\Omega}$ hides log factors in m and K is a universal constant.

These parameters have been generalized to the online learning setting using binary trees. A $\mathcal{X}$-valued tree $\mathbf{x}$ of depth n is a rooted complete binary tree with nodes labeled by elements of $\mathcal{X}$. We identify the tree $\mathbf{x}$ with the sequence $(\mathbf{x}_1, \ldots, \mathbf{x}_m)$ of labeling functions $\mathbf{x}_i : \{\pm 1\}^{i-1} \mapsto \mathcal{X}$ which provide the labels for each node. Here, $\mathbf{x}_1 \in \mathcal{X}$ is the label for the *root* of the tree, while $\mathbf{x}_i$ for $i > 1$ is the label of the node obtained by following the path of length $i-1$ from the root, with $+1$ indicating 'right' and -1 indicating 'left'. A *path* of length m is given by the sequence $\sigma = (\sigma_1, \ldots, \sigma_m) \in \{\pm 1\}^m$. For brevity, we shall often write $\mathbf{z}_t(\sigma)$, but it is understood that $\mathbf{z}_t$ only depends only on the prefix $(\sigma_1, \ldots, \sigma_{t-1})$ of σ.

Definition 2.11. The *sequential Rademacher complexity* of $\mathcal{F}$ on a $\mathcal{X}$-valued tree $\mathbf{x}$ is defined to be

$$\hat{\mathsf{srad}}_m(\mathcal{F}; \mathbf{x}) = \mathbb{E}_\sigma\left[\sup_{f\in\mathcal{F}}\left(\frac{1}{m}\sum_{i=1}^m \sigma_i f(\mathbf{x}_i(\sigma))\right)\right]$$

Definition 2.12. The *sequential Rademacher complexity* of $\mathcal{F}$ is defined as

$$\mathsf{srad}_m(\mathcal{F}) = \sup_{\mathbf{x}}[\hat{\mathsf{srad}}_m(\mathcal{F}; \mathbf{x})]$$

Concentration. The Azuma-Hoeffding Inequality is an essential component of showing concentration in online algorithms and online-to-offline conversions.

Lemma 2.3 (Azuma-Hoeffding's Inequality)**.** If $X_1, \ldots, X_T$ is a martingale difference sequence, and for every t, with probability 1, $|X_t| \leq M$. Then with probability $1-\delta$,

$$\left|\sum_{t=1}^T X_t\right| \leq M\sqrt{2T\ln\frac{2}{\delta}}.$$

3 Proper Calibration

In this section, we introduce the notion of proper calibration (Definition 3.1). We show that proper calibration is actually a restatement of the notion of Decision OI, used by Gopalan et al. (2023) to achieve omniprediction. Because proper calibration is defined in terms of proper loss functions, we leverage a characterization of proper losses by the V-shaped losses (Li et al., 2022; Kleinberg et al., 2023) to give an efficient strategy for achieving proper calibration.

Defining Proper Calibration. We define proper calibration as an instance of weighted calibration. In particular, we use a class of weight functions $\mathcal{W}_{\text{proper}}$ associated with proper scoring rules, where each weight function $w \in \mathcal{W}_{\text{proper}}$ corresponds to the discrete derivative (Definition 2.6) of a proper loss function. Concretely, $\mathcal{W}_{\text{proper}}$ is defined as the following set of weight functions.

$$\mathcal{W}_{\text{proper}} = \{\Delta\ell : \ell \in \mathcal{L}_{\text{proper}}\}$$

Proper calibration ensures that the predictions, weighted by the discrete derivatives of any proper loss, do not correlate with the residual prediction error.

Definition 3.1 (Proper Calibration)**.** For a sequence of T predictions $\mathbf{p}$ and context, outcome pairs $\mathbf{x}, \mathbf{y}$, the proper calibration error is given by $\mathcal{W}_{\text{proper}}$-CalErr$(\mathbf{p}, \mathbf{x}, \mathbf{y})$ i.e

$$\text{PCalErr}(\mathbf{p}, \mathbf{x}, \mathbf{y}) = \sup_{w \in \mathcal{W}_{\text{proper}}} \left| \sum_{t=1}^{T} w(p_t(x_t))(y_t - p_t(x_t)) \right|$$

When $\mathbf{x}, \mathbf{y}$ are drawn from a fixed distribution $\mathcal{D}$, we shall refer to the expected proper calibration error over the distribution by $\text{PCalErr}_{\mathcal{D}}(\mathbf{p})$ defined as follows:

$$\text{PCalErr}_{\mathcal{D}}(\mathbf{p}) = \sup_{w \in \mathcal{W}_{\text{proper}}} \left| \mathop{\mathbf{E}}_{\mathcal{D}} \left[w(p(x))(y - p(x)) \right] \right|$$

In our understanding, proper calibration is a novel notion of calibration not known to be implied by other notions, other than full ℓ_1-calibration. We provide a more complete comparison of proper calibration to prior notions of calibration in Appendix A. Next, we describe why proper calibration is natural and useful in the context of omniprediction.

3.1 Omniprediction via Proper Calibration and Multiaccuracy

To motivate proper calibration, we recall the notion of Decision OI introduced by Gopalan et al. (2023). We restate their definition and extend it to include the sequential prediction setting.

Definition 3.2 (Decision OI)**.** Fix a collection of loss functions $\mathcal{L}$. For a predictor $p : \mathcal{X} \to [0, 1]$, the Decision OI error over a distribution $\mathcal{D}$ is given by $\mathcal{L}$-DecOIErr$_{\mathcal{D}}(p)$.

$$\mathcal{L}\text{-DecOIErr}_{\mathcal{D}}(p) = \sup_{\ell \in \mathcal{L}} \left| \mathop{\mathbf{E}}_{x,y \sim \mathcal{D}} [\ell(k_\ell \circ p(x), y)] - \mathop{\mathbf{E}}_{\substack{x \sim \mathcal{D} \\ \tilde{y} \sim \text{Ber}(p(x))}} [\ell(k_\ell \circ p(x), \tilde{y})] \right|$$

For a sequence of T predictions $\mathbf{p}$ and context, outcome pairs $\mathbf{x}, \mathbf{y}$, the Decision OI error is given by $\mathcal{L}$-DecOIErr$(\mathbf{p}, \mathbf{x}, \mathbf{y})$.

$$\mathcal{L}\text{-DecOIErr}(\mathbf{p}, \mathbf{x}, \mathbf{y}) = \sup_{\ell \in \mathcal{L}} \left| \sum_{t=1}^{T} (\ell(k_\ell \circ p_t(x_t), y_t) - \mathop{\mathbf{E}}_{\tilde{y} \sim \text{Ber}(p_t(x_t))} \ell(k_\ell \circ p_t(x_t), \tilde{y})) \right|$$

As in all notions of outcome indistinguishability, Decision OI compares the expected value of some test on outcomes sampled from the "real" world (i.e., y_t) versus the value on outcomes sampled from a "modeled" world, where outcomes are sampled according to our predictions $\tilde{y} \sim \text{Ber}(p_t(x_t))$.

Gopalan et al. (2023) showed that Decision OI plus a certain multiaccuracy condition suffice for omniprediction; they use full ℓ_1-calibration to achieve Decision OI. We show that, for any loss class $\mathcal{L}$, proper calibration suffices to imply Decision OI. In fact, Proper Calibration is equivalent to Decision OI for the class of all loss function $\mathcal{L}_{\text{all}} = \{\ell : [0, 1] \to [-1, 1]\}$.

Theorem 3.1. For any predictor $p : \mathcal{X} \to [0, 1]$ and distribution $\mathcal{D}$,

$$\mathcal{L}_{\text{all}}\text{-DecOIErr}_{\mathcal{D}}(p) = \text{PCalErr}_{\mathcal{D}}(p).$$

For every sequence of T predictions $\mathbf{p}$ and context, outcome pairs $\mathbf{x}, \mathbf{y}$

$$\mathcal{L}_{\text{all}}\text{-DecOIErr}(\mathbf{p}, \mathbf{x}, \mathbf{y}) = \text{PCalErr}(\mathbf{p}, \mathbf{x}, \mathbf{y}).$$

Proof. In the distributional setting, we rewrite the definition of Decision OI. We write the difference of losses in terms of the difference of their discrete derivatives,

$$\begin{aligned} \mathcal{L}_{\text{all}}\text{-DecOIErr}_{\mathcal{D}}(p) &= \sup_{\ell \in \mathcal{L}_{\text{all}}} \left| \mathop{\mathbf{E}}_{x,y \sim \mathcal{D}} [\ell(k_\ell \circ p(x), y)] - \mathop{\mathbf{E}}_{\substack{x \sim \mathcal{D} \\ \tilde{y} \sim \text{Ber}(p(x))}} [\ell(k_\ell \circ p(x), \tilde{y})] \right| \\ &= \sup_{\ell \in \mathcal{L}_{\text{all}}} \left| \mathop{\mathbf{E}}_{x,y \sim \mathcal{D}} [y \cdot \Delta\ell(k_\ell \circ p(x)) - p(x) \cdot \Delta\ell(k_\ell \circ p(x))] \right| \end{aligned}$$

where the equality follows by the fact that, in the modeled world, $\mathbf{E}[\tilde{y}|x] = p(x)$ and the definition of $\Delta\ell$. Then, we can combine terms and consider the supremum.

$$\sup_{\ell \in \mathcal{L}_{\text{all}}} \left| \mathop{\mathbf{E}}_{x,y \sim \mathcal{D}} [\Delta\ell(k_\ell \circ p(x)) \cdot (y - p(x))] \right| = \sup_{\ell \in \mathcal{L}_{\text{proper}}} \left| \mathop{\mathbf{E}}_{x,y \sim \mathcal{D}} [\Delta\ell(p(x)) \cdot (y - p(x))] \right|$$

The final equality, here, follows from the fact that for any loss, ℓ composed with the optimal post-processing k_ℓ, can be viewed as a proper loss. Formally, for any $\ell : [0, 1] \to [-1, 1]$, there exists a proper loss $\ell' \in \mathcal{L}_{\text{proper}}$ (namely, $\ell'(p, y) = \ell(k_\ell(p), y)$) such that $\Delta\ell(k_\ell(p)) = \Delta\ell'(p)$, for all $p \in [0, 1]$.

Note that the same argument works in the sequential setting, establishing the analogous equality. □

Using this view on Proper Calibration, we can immediately apply the Loss OI framework to achieve omnipredictors, with proper calibration replacing Decision OI / calibration.

Lemma 3.2. For a sequence of T predictions $\mathbf{p}$ and context, outcome pairs $\mathbf{x}, \mathbf{y}$,

$$(\mathcal{L}, \mathcal{H})\text{-OmniRegret}(\mathbf{p}, \mathbf{x}, \mathbf{y}) \leq \text{PCalErr}(\mathbf{p}, \mathbf{x}, \mathbf{y}) + (\Delta\mathcal{L} \circ \mathcal{H})\text{-MAErr}(\mathbf{p}, \mathbf{x}, \mathbf{y})$$

The proof of lemma mirrors that of Proposition 4.5 in Gopalan et al. (2023) in that it explicitly follows the Loss OI framework introduced in the paper. For completeness and self-containment, we give a proof of the statement in the sequential setting.

Proof of Lemma 3.2. Recall that

$$(\mathcal{L},\mathcal{H})\text{-OmniRegret}(\mathbf{p},\mathbf{x},\mathbf{y}) = \max_{\{h\in\mathcal{H},\ell\in\mathcal{L}\}} \sum_{t=1}^{T} \ell(k_\ell(p_t(x_t)),y_t) - \ell(h(x_t),y_t)$$

For a fixed timestep $t \in [T]$, we expand the inner expression as follows:

$$\begin{aligned}
&\ell(k_\ell(p_t(x_t)),y_t) - \ell(h(x_t),y_t) && (11)\\
&\le \ell(k_\ell(p_t(x_t)),y_t) + \mathop{\mathbf{E}}_{\tilde{y}\sim\mathrm{Ber}(p_t(x_t))}[\ell(h(x_t),\tilde{y})] - \mathop{\mathbf{E}}_{\tilde{y}\sim\mathrm{Ber}(p_t(x_t))}[\ell(k_\ell(p_t(x_t)),\tilde{y})] - \ell(h(x_t),y_t) && (12)\\
&= \left[\ell(k_\ell(p_t(x_t)),y_t) - \mathop{\mathbf{E}}_{\tilde{y}\sim\mathrm{Ber}(p_t(x_t))}[\ell(k_\ell(p_t(x_t)),\tilde{y})]\right] + \left[\mathop{\mathbf{E}}_{\tilde{y}\sim\mathrm{Ber}(p_t(x_t))}[\ell(h(x_t),\tilde{y})] - \ell(h(x_t),y_t)\right] && (13)
\end{aligned}$$

where the second line follows from the definition of k_ℓ. Now we simplify both terms separately. For the first term, we have

$$\begin{aligned}
&\ell(k_\ell(p_t(x_t)),y_t) - \mathop{\mathbf{E}}_{\tilde{y}\sim\mathrm{Ber}(p_t(x_t))}[\ell(k_\ell(p_t(x_t)),\tilde{y})] && (14)\\
&= [y_t\Delta\ell(k_\ell(p_t(x_t))) + \ell(k_\ell(p_t(x_t)),0)] - [p_t\Delta\ell(k_\ell(p_t(x_t))) + \ell(k_\ell(p_t(x_t))),0)] && (15)\\
&= (y_t - p_t(x_t))\Delta\ell(k_\ell(p_t(x_t))) && (16)
\end{aligned}$$

where line (6) uses the fact that $\ell(p,y) = y(\ell(p,1)-\ell(p,0))+\ell(p,0)$ for $y\in\{0,1\}$ Similarly, for the second term, we have

$$\begin{aligned}
&\mathop{\mathbf{E}}_{\tilde{y}\sim\mathrm{Ber}(p_t(x_t))}[\ell(h(x_t),\tilde{y})] - \ell(h(x_t),y_t) && (17)\\
&= [p_t(x_t)\Delta\ell(h(x_t)) + \ell(h(x_t),0)] - [y_t\Delta\ell(h(x_t)) + \ell(h(x_t),0)] && (18)\\
&= (p_t(x_t) - y_t)\Delta\ell(h(x_t)) && (19)
\end{aligned}$$

Plugging these simplifications back into the regret term, we obtain

$$\begin{aligned}
(\mathcal{L},\mathcal{H})\text{-OmniRegret}(\mathbf{p},\mathbf{x},\mathbf{y}) &\le \max_{\{h\in\mathcal{H},\ell\in\mathcal{L}\}} \sum_{t=1}^{T}(y_t - p_t(x_t))\Delta\ell(k_\ell(p_t(x_t))) + (p_t(x_t)-y_t)\Delta\ell(h(x_t)) && (20)\\
&\le \max_{\{\ell\in\mathcal{L}\}}\left|\sum_{t=1}^{T}(y_t - p_t(x_t))\Delta\ell(k_\ell(p_t(x_t)))\right| + \max_{\{h\in\mathcal{H},\ell\in\mathcal{L}\}}\left|\sum_{t=1}^{T}(p_t(x_t)-y_t)\Delta\ell(h(x_t))\right| && (21)\\
&\le \max_{\{\ell\in\mathcal{L}_{\text{proper}}\}}\left|\sum_{t=1}^{T}(y_t - p_t(x_t))\Delta\ell(p_t(x_t))\right| + \max_{\{h\in\mathcal{H},\ell\in\mathcal{L}\}}\left|\sum_{t=1}^{T}(p_t(x_t)-y_t)\Delta\ell(h(x_t))\right| && (22)\\
&\le \text{PCalErr}(\mathbf{p},\mathbf{x},\mathbf{y}) + (\Delta\mathcal{L}\circ\mathcal{H})\text{-MAErr}(\mathbf{p},\mathbf{x},\mathbf{y}) && (23)
\end{aligned}$$

where line (22) follows from the fact that $\Delta\ell$ composed with its k_ℓ function corresponds to some $\Delta\ell_{\text{proper}}$ for some proper loss $\ell_{\text{proper}} \in \mathcal{L}_{\text{proper}}$. □

Naturally, proper calibration can be incorporated into the original algorithms for learning omnipredictors via Loss OI. In Appendix C, we describe an adaptation of the boosting-style algorithm of Gopalan et al. (2023) that outputs an omnipredictor using a multiplicative factor of $1/\varepsilon^6$ fewer samples than the original algorithm based on ℓ_1-calibration.

3.2 Approximating Proper Calibration with Weighted Calibration over Thresholds

To achieve calibration for all proper weight functions, we will take advantage of the basis decomposition of proper scoring losses in (Li et al., 2022; Kleinberg et al., 2023). To this end, we introduce a notion of calibration, weighted by threshold functions that utilizes the basis functions as weight functions to approximate proper calibration effectively. Concretely, we consider the following characterization of proper losses.

Lemma 3.3 (V-shaped proper losses (Li et al., 2022; Kleinberg et al., 2023))**.** For $v \in [0,1]$, define the proper loss $\ell_v(p,y) = (y-v)\text{sgn}(v-p)$. Then every $\ell \in \mathcal{L}_{\text{proper}}$ can be expressed as a convex combination of these v-shaped proper losses. That is, for every $\ell \in \mathcal{L}_{\text{proper}}$, there exists nonnegative coefficients $c_v(\ell)$ such that $\int_0^1 c_v(\ell)dv \le 2$ and

$$\ell(p,y) = \int_0^1 c_v(\ell)\ell_v(p,y)dv$$

Immediately, this characterization also gives a characterization of the discrete derivatives of proper losses.

Corollary 3.4. For every $\ell \in \mathcal{L}_{\text{proper}}$, there exists nonnegative coefficients $c_v(\ell)$ such that $\int_0^1 c_v(\ell)dv \le 2$ and

$$\Delta\ell(p) = \int_0^1 c_v(\ell)\Delta\ell_v(p)dv$$

where $\Delta\ell_v(p) = \text{sgn}(v-p)$ is a {-1,1} threshold function at v.

In the Lemma 3.5, we show that proper calibration and calibration weighted by threshold functions are within a constant factor of each other. Specifically, we let $\mathcal{W}_{\text{Th}}$ denote the set of weight functions associated with V-shaped proper scoring rules, where each weight function w corresponds to the discrete derivative of a V-shaped proper loss function i.e $\mathcal{W}_{\text{Th}} = \{\text{sgn}(v-p) : v \in [0,1]\}$. Then, we obtain the following tight approximation of the proper calibration error.

Lemma 3.5. For a sequence of T predictions $\mathbf{p}$ and context, outcome pairs $\mathbf{x}, \mathbf{y}$,

$$\mathcal{W}_{\text{Th}}\text{-CalErr}(\mathbf{p},\mathbf{x},\mathbf{y}) \le \text{PCalErr}(\mathbf{p},\mathbf{x},\mathbf{y}) \le 2\mathcal{W}_{\text{Th}}\text{-CalErr}(\mathbf{p},\mathbf{x},\mathbf{y})$$

Proof. We wish to show that

$$\mathcal{W}_{\text{Th}}\text{-CalErr}(\mathbf{p},\mathbf{x},\mathbf{y}) \le \mathcal{W}_{\text{proper}}\text{-CalErr}(\mathbf{p},\mathbf{x},\mathbf{y}) \le 2\mathcal{W}_{\text{Th}}\text{-CalErr}(\mathbf{p},\mathbf{x},\mathbf{y})$$

By Corollary 3.4, we know that for every $\ell \in \mathcal{L}_{\text{proper}}$, there exists nonnegative coefficients $c_v(\ell)$ such that $\int_0^1 c_v(\ell)dv \le 2$

$$\Delta\ell(p) = \int_0^1 c_v(\ell)\Delta\ell_v(p)dv$$

$$\max_{w\in\mathcal{W}_{\text{proper}}}\left|\sum_{t=1}^{T} w(p_t(x_t))(y_t - p_t(x_t))\right| = \max_{\ell\in\mathcal{L}_{\text{proper}}}\left|\sum_{t=1}^{T}\Delta\ell(p_t(x_t))(y_t - p_t(x_t))\right| \tag{24}$$

$$= \max_{\ell\in\mathcal{L}_{\text{proper}}}\left|\sum_{t=1}^{T}\left(\int_0^1 c_v(\ell)\Delta\ell_v(p_t(x_t))dv\right)(y_t - p_t(x))\right| \quad \text{(Corollary 3.4)}$$

$$= \max_{\ell\in\mathcal{L}_{\text{proper}}}\left|\int_0^1 c_v(\ell)\left(\sum_{t=1}^{T}\Delta\ell_v(p_t(x_t))(y_t - p_t(x))\right)dv\right| \tag{25}$$

$$\leq \max_{\ell\in\mathcal{L}_{\text{proper}}}\left|\int_0^1 c_v(\ell)\sup_{v\in[0,1]}\left(\sum_{t=1}^{T}\Delta\ell_v(p_t(x_t))(y_t - p_t(x))\right)dv\right| \tag{26}$$

$$\leq \max_{\ell\in\mathcal{L}_{\text{proper}}}\left|\int_0^1 c_v(\ell)dv\right|\sup_{v\in[0,1]}\left|\sum_{t=1}^{T}\Delta\ell_v(p_t(x_t))(y_t - p_t(x))\right| \tag{27}$$

$$\leq 2\sup_{v\in[0,1]}\left|\sum_{t=1}^{T}\Delta\ell_v(p_t(x_t))(y_t - p_t(x))\right| \tag{28}$$

Thus,

$$\mathcal{W}_{\text{proper}}\text{-CalErr}(\mathbf{p},\mathbf{x},\mathbf{y}) \leq 2\mathcal{W}_{\text{Th}}\text{-CalErr}(\mathbf{p},\mathbf{x},\mathbf{y})$$

The first inequality $\mathcal{W}_{\text{Th}}\text{-CalErr}(\mathbf{p},\mathbf{x},\mathbf{y}) \leq \mathcal{W}_{\text{proper}}\text{-CalErr}(\mathbf{p},\mathbf{x},\mathbf{y})$ follows from the fact that $\mathcal{W}_{\text{Th}} \subset \mathcal{W}_{\text{proper}}$ □

Lemma 3.5 bounds the proper calibration in terms of a notion of weighted calibration with an uncountably infinite collection of weight functions $\mathcal{W}_{\text{Th}}$. Towards a practically-realizable algorithm for proper calibration, we show that for sufficiently discretized predictions, weighted calibration with respect to a discretized collection of thresholds suffices to bound the proper calibration error. Let $\mathcal{W}_{\text{Th}}^{\gamma}$ denote an γ-discretization of $\mathcal{W}_{\text{Th}}$. That is, $\mathcal{W}_{\text{Th}}^{\gamma} = \{\text{sgn}(v - p) : v \in \{0, \gamma, 2\gamma, \ldots, 1\}\}$.

Lemma 3.6 (γ-discretized thresholds)**.** For a sequence of T predictions $\mathbf{p}$ with values in $\{0, \gamma, 2\gamma, \ldots, 1\}$ and context, outcome pairs $\mathbf{x}, \mathbf{y}$,

$$\mathcal{W}_{\text{Th}}\text{-CalErr}(\mathbf{p},\mathbf{x},\mathbf{y}) = \mathcal{W}_{\text{Th}}^{\gamma}\text{-CalErr}(\mathbf{p},\mathbf{x},\mathbf{y})$$

Proof of Lemma 3.6. Recall that

$$\mathcal{W}_{\text{Th}}\text{-CalErr}(\mathbf{p},\mathbf{x},\mathbf{y}) = \max_{v\in[0,1]}\left|\sum_{t=1}^{T}\text{sgn}(v - p_t(x_t))\cdot(y_t - p_t(x_t))\right|.$$

Since the predictions $p_t(x_t)$ take values only in $\{0, \gamma, \ldots, 1\}$, the sign function $\text{sgn}(v - p_t(x_t))$ can only change at points in $\{0, \gamma, \ldots, 1\}$. Hence, for any $v \in [0, 1]$, there exists some $v' \in \{0, \gamma, \ldots, 1\}$ such that:

$$\text{sgn}(v - p_t(x_t)) = \text{sgn}(v' - p_t(x_t)).$$

This implies

$$\sum_{t=1}^{T}\text{sgn}(v - p_t(x_t))\cdot(y_t - p_t(x_t)) = \sum_{t=1}^{T}\text{sgn}(v' - p_t(x_t))\cdot(y_t - p_t(x_t)).$$

Therefore

$$\max_{v\in[0,1]}\left|\sum_{t=1}^{T}\operatorname{sgn}(v-p_t(x_t))\cdot(y_t-p_t(x_t))\right| = \max_{v'\in\{0,\gamma,\ldots,1\}}\left|\sum_{t=1}^{T}\operatorname{sgn}(v'-p_t(x_t))\cdot(y_t-p_t(x_t))\right|.$$

Thus

$$\mathcal{W}_{\text{Th}}\text{-CalErr}(\mathbf{p},\mathbf{x},\mathbf{y}) = \mathcal{W}^{\gamma}_{\text{Th}}\text{-CalErr}(\mathbf{p},\mathbf{x},\mathbf{y}).$$

□

3.3 Algorithm for Proper Calibration in the Online Setting

In this subsection, we present an algorithm that guarantees proper calibration in the online setting at the rate of $O(\sqrt{T\log T})$. Algorithm 1 achieves this by ensuring proper calibration by auditing with threshold weight functions per Lemma 3.5. Moreover, the rate of our algorithm is nearly tight; no algorithm can guarantee expected threshold calibration error at a rate of $\Omega(\sqrt{T})$. This lower bound is unsurprising, but follows by Appendix A.2, where we show that U-calibration lower bounds a multiplicative factor of proper calibration.

Overview of Algorithm 1: The algorithm is based on Blackwell's Approachability Theorem. We define a two player game where the adversary player selects $z_t = (x_t, y_t) \in \mathcal{X}\times\{0,1\}$ and the learner selects $p_t : \mathcal{X}\to[1/T]$. Both players are allowed to play randomized strategies but since the learner observes x_t, we can simplify things and only consider $y_t \in \{0,1\}$ and $p_t = p_t(x_t)$. We design the payoff vector of this game to reflect our objective of threshold calibration. That is, define

$$u_{v,s}(p_t,y_t) = s(y_t-p_t)\operatorname{sgn}(v-p_t) \quad \text{for } v\in[1/T], s\in\{+,-\}$$

Observe that after T rounds of interaction, $\mathcal{W}_{\text{Th}}\text{-CalErr}(\mathbf{p},\mathbf{x},\mathbf{y}) = \max_{v,s}\sum_{t\in[T]}u_{v,s}(p_t,z_t)$. Therefore, we design the learner's target set to be the set of all vectors u with coordinates less than $1/T$.

We use exponential weights update method in Line 2 to generate sequence of halfspaces w^t with coordinates for every $v\in[1/T], s\in\{+,-\}$. Given a halfspace w^t, the algorithm computes the function $f(q)$ defined in Line 3 and the algorithm chooses the distribution to sample p_t using the strategy described in Line 4 to Line 10.

In all, we establish the following regret bound.

Theorem 3.7 (Proper Calibration Upper Bound)**.** Algorithm 1 guarantees expected proper calibration error of $O\left(\sqrt{T\ln T}\right)$

Formally, we leverage the following lemmas that give guarantees on the Blackwell optimal response (in Line 4 to Line 10) and the dual player's strategy (in Line 2).

Lemma 3.8 (Halfspace Approachability)**.** Given a halfspace w, the strategy described in Line 4 to Line 10 outputs a distribution p_t over $[1/T]$ such that $\mathbf{E}_{p_t}[\langle w, u(p_t,z_t)\rangle] \le 1/T$ for any choice of z_t

Proof. We consider the cases in the strategy separately:

Algorithm 1 Proper Calibration

Input: Sequence of samples $\{y_1, \ldots, y_T\}$

Output: Sequence of (randomized) predictors $p_1, \ldots, p_T$

1: **for** each $t \in [T]$ **do**

2: Let $w^t_{v,s} := \frac{\exp\left(\eta \sum_{i=1}^{t-1} u_{v,s}(p_i, z_i)\right)}{\sum_{v',s'} \exp\left(\eta \sum_{i=1}^{t-1} u_{v',s'}(p_i, z_i)\right)}$ for all $v \in [1/T]$, $s \in \{+, -\}$

3: Compute

$$f(q) = \sum_{v,s} w^t_{v,s} \cdot s \cdot \mathrm{sgn}(v - q)$$

4: **if** $f(0) \le 0$ **then**

5: Predict $p_t = 0$

6: **else if** $f(1) > 0$ **then**

7: Predict $p_t = 1$

8: **else**

9: Find adjacent probabilities q_i, q_{i+1} such that $f(q_i) \cdot f(q_{i+1}) \le 0$

10: Predict $p_t = q_i$ with prob $\frac{|f(q_{i+1})|}{|f(q_i)|+|f(q_{i+1})|}$ and $p_t = q_{i+1}$ with prob $\frac{|f(q_i)|}{|f(q_i)|+|f(q_{i+1})|}$

11: Observe x_t, predict p_t, and then observe y_t

Case 1: If $f(0) \le 0$, predict $p_t = 0$. Then for $v \in [1/T]$, $s \in \{+, -\}$,

$$w_{v,s} u_{v,s}(p_t, z_t) = s(y_t - p_t)\mathrm{sgn}(v - p_t) w_{v,s} = s y_t w_{v,s}$$

Summing over values of v, s and applying the definition of f in Line 3, we get

$$\langle w, u(p_t, z_t)\rangle = y_t f(0) \le 0 \quad \text{for any choice of } y \in \{0, 1\}$$

Case 2: If $f(1) > 0$, predict $p_t = 1$. Then for $v \in [1/T]$, $s \in \{+, -\}$,

$$w_{v,s} u_{v,s}(p_t, z_t) = s(y_t - p_t)\mathrm{sgn}(v - p_t) w_{v,s} = s y_t w_{v,s}$$

Summing over values of v, s and applying the definition of f in Line 3, we get

$$\langle w, u(p_t, z_t)\rangle = (y_t - 1) f(1) \le 0 \quad \text{for any choice of } y \in \{0, 1\}$$

Case 3: Find adjacent probabilities q_i, q_{i+1} such that $f(q_i) \cdot f(q_{i+1}) \le 0$.
We wish to bound $\mathbf{E}_{p_t}[\langle w, u(p_t, y)\rangle]$ for both possible outcomes $y \in \{0, 1\}$. Let's fix v, s and compute $\mathbf{E_p}[w_{v,s} u_{v,s}]$

$$\frac{|f(q_{i+1})|}{|f(q_i)| + |f(q_{i+1})|} \left[s(y - q_i)\mathrm{sgn}(v - q_i) w_{v,s}\right] + \frac{|f(q_i)|}{|f(q_i)| + |f(q_{i+1})|} \left[s(y - q_{i+1})\mathrm{sgn}(v - q_{i+1}) w_{v,s}\right]$$

Summing over all pairs of (v, s) and applying the definition of f in Line 3, we obtain

$$\frac{|f(q_{i+1})|}{|f(q_i)| + |f(q_{i+1})|} \left[(y - q_i) f(q_i)\right] + \frac{|f(q_i)|}{|f(q_i)| + |f(q_{i+1})|} \left[(y - q_{i+1}) f(q_{i+1})\right] \tag{29}$$

$$\le \frac{|f(q_i)||f(q_{i+1})|}{|f(q_i)| + |f(q_{i+1})|}(q_{i+1} - q_i) \tag{30}$$

$$\le 1/T \qquad \text{(since } f(q) \in [-1, 1]\text{)}$$

□

Lemma 3.9 (Exponential Weight Updates Arora et al. (2012))**.** The exponential weight updates in Line 2 provide a sequence of vectors w^t such that

$$\max_{w:\|w\|_1=1} \left\langle w, \sum_{t\in[T]} u(p_t, z_t) \right\rangle \leq \sum_{t=1}^{T} \langle w^t, u(p_t, z_t)\rangle + O\left(\sqrt{T \ln T}\right)$$

With these lemmas in place, we can prove Theorem 3.7.

Proof of Theorem 3.7. We wish to bound the expected threshold calibration error, that is, $\mathbf{E}_{\mathbf{p}}\left[\max_{v,s} \sum_{t\in[T]} u_{v,s}(p_t, z_t)\right]$ where the expectation is over the randomness in the sampling of p_t. Note that this is the same as $\mathbf{E}_{\mathbf{p}}\left[\max_{w:\|w\|_1=1} \left\langle w, \sum_{t\in[T]} u(p_t, z_t)\right\rangle\right]$.

$$\begin{aligned}
\mathop{\mathbf{E}}_{\mathbf{p}}\left[\max_{w:\|w\|_1=1} \left\langle w, \sum_{t\in[T]} u(p_t, z_t)\right\rangle\right] &\leq \mathop{\mathbf{E}}_{\mathbf{p}}\left[\sum_{t=1}^{T} \langle w^t, u(p_t, z_t)\rangle\right] + O\left(\sqrt{T \ln T}\right) && \text{(by Lemma 3.9)} \\
&= \sum_{t=1}^{T} \mathop{\mathbf{E}}_{\mathbf{p}} \langle w^t, u(p_t, z_t)\rangle + O\left(\sqrt{T \ln T}\right) && \text{(by linearity of expectation)} \\
&= \sum_{t=1}^{T} \mathop{\mathbf{E}}_{w^t}\left[\mathop{\mathbf{E}}_{p_t} \langle w^t, u(p_t, z_t)\rangle \,\middle|\, w^t\right] + O\left(\sqrt{T \ln T}\right) \\
& && \text{(by law of iterated expectations)} \\
&\leq \left(\sum_{t=1}^{T} \tfrac{1}{T}\right) + O\left(\sqrt{T \ln T}\right) && \text{(by Lemma 3.8)} \\
&\leq O\left(\sqrt{T \ln T}\right)
\end{aligned}$$

□

In all, proper calibration is an intermediate notion: strong enough to imply omniprediction via the Loss OI framework, but weak enough that it can be achieved asymptotically efficiently compared to ℓ_1-calibration.

3.4 Augmented Proper Calibration

To aid with our goal of achieving proper calibration and multiaccuracy simultaneously, we introduce the following augmented proper calibration problem. In the augmented proper calibration problem we are given a sequence of samples $(x_1, y_1), \ldots, (x_T, y_T)$ and hypothesis functions $q_1, \ldots, q_T : \mathcal{X} \to [-1, 1]$ and the goal is to output a sequence of predictors $p_1, \ldots, p_T : \mathcal{X} \to [0, 1]$ that is proper calibrated i.e PCalErr($\mathbf{p}, \mathbf{x}, \mathbf{y}$) is sublinear and satisfy the following sequential multiaccuracy guarantee $\sum_{t=1}^{T} q_t(x_t)(y_t - p_t(x_t)) \in o(T)$ with respect to the sequence of input hypothesis functions $q_1, \ldots, q_T : \mathcal{X} \to [-1, 1]$. To solve this problem we present the algorithm, Augmented Proper Calibration (APCAL), and state its performance guarantees in both the sequential and distributional settings.

Proposition 3.10 (Augmented Proper Calibrator)**.** Given an online sequence of samples $(x_1, y_1), \ldots, (x_T, y_T)$ and hypothesis functions $q_1, \ldots, q_T$, Algorithm 2 outputs a sequence of predictors $p_1, \ldots, p_T$ such that with probability at least $1 - \delta$,

Algorithm 2 Augmented Proper Calibration

procedure APCAL(T)

Input: Sequence of samples $(x_1, y_1), \ldots, (x_T, y_T)$

Input: Sequence of hypothesis functions $q_1, \ldots, q_T : \mathcal{X} \to [-1, 1]$

Output: Sequence of predictors $p_1, \ldots, p_T : \mathcal{X} \to [0, 1]$

Initialize weights $w^1_{v,s} = \frac{1}{2T+3}$ for $v \in [1/T]$, $s \in \{+, -\}$ ▷ for proper calibration constraints

Initialize weight: $w^1_{\text{sma}} = \frac{1}{2T+3}$ ▷ for sequential multiaccuracy constraint

Set $\eta = \sqrt{\log(T)/T}$ ▷ learning rate for exponential weight updates

for $t = 1$ to T **do**

Receive q_t

Compute

$$f(x, j) = w^t_{\text{sma}} \cdot q_t(x) + \sum_{v,s} w^t_{v,s} \cdot s \cdot \text{Th}_v(j)$$

Compute predictor $p_t(x)$ as follows:

if $f(x, 0) \le 0$ **then**

Predict $p_t(x) = 0$

else if $f(x, 1) > 0$ **then**

Predict $p_t(x) = 1$

else

Find adjacent probabilities $\frac{i}{T}, \frac{i+1}{T}$ such that $f\left(x, \frac{i}{T}\right) \cdot f\left(x, \frac{i+1}{T}\right) \le 0$

Predict $p_t(x) = \frac{i}{T}$ with prob $\frac{|f(x,(i+1)/T)|}{|f(x,i/T)|+|f(x,(i+1)/T))|}$ and $p_t = \frac{i+1}{T}$ with prob $\frac{|f(x,i/T)|}{|f(x,i/T)|+|f(x_t,(i+1)/T)|}$

Return p_t

Observe (x_t, y_t)

Perform exponential weight updates with losses $s \cdot (y_t - p_t(x_t)) \cdot \text{Th}_v(p_t(x_t))$ for each v, s and $(y_t - p_t(x_t)) \cdot q_t(x_t)$ for the sequential multiaccuracy constraint:

$$W^t = e^{-\eta \cdot (y_t - p_t(x_t)) \cdot q_t(x_t)} w^t_{\text{sma}} + \sum_{v,s} e^{-\eta \cdot s \cdot (y_t - p_t(x_t)) \cdot \text{Th}_v(p_t(x_t))} w^t_{v,s}$$

$$w^{t+1}_{\text{sma}} = e^{-\eta \cdot (y_t - p_t(x_t)) \cdot q_t(x_t)} w^t_{\text{sma}} \,/\, W^t$$

$$w^{t+1}_{v,s} = e^{-\eta \cdot s \cdot (y_t - p_t(x_t)) \cdot \text{Th}_v(p_t(x_t))} w^t_{v,s} \,/\, W^t$$

1. $\text{PCalErr}(\mathbf{p}, \mathbf{x}, \mathbf{y}) \le O\left(\sqrt{T \ln T/\delta}\right)$
2. $\sum_{t=1}^{T} q_t(x_t)(y_t - p_t(x_t)) \le O\left(\sqrt{T \ln T/\delta}\right)$.

If the samples $\{(x_t, y_t)\}_{t=1}^{T}$ are drawn i.i.d. from a distribution $\mathcal{D}$ (while the functions $q_1, \ldots, q_T$ may still be specified online by an adaptive adversary) then the uniform distribution $\hat{\mathbf{p}} = \text{unif}(p_1, \ldots, p_T)$ satisfies the following bounds with probability at least $1 - \delta$.

1. $\text{PCalErr}_{\mathcal{D}}(\hat{\mathbf{p}}) \le O\left(\sqrt{\frac{\ln(T/\delta)}{T}}\right)$

2. $\frac{1}{T}\sum_{t=1}^{T}\mathbf{E}_{(x,y)\sim\mathcal{D}}[q_t(x)(y-p_t(x))] \le O\left(\sqrt{\frac{\ln(T/\delta)}{T}}\right)$.

The algorithm runs in poly(T) timesteps per iteration and each p_t is simply a remapping of q_t i.e $p_t(x) = u(q_t(x))$ for some (random) function $u : [-1, 1] \to [1/T]$.

Overview of Algorithm 2: The algorithm follows the design ideas of Algorithm 1. We consider a two player game that combines the objectives of proper calibration and sequential multiaccuracy guarantee. The adversary selects $z_t = (x_t, y_t) \in \mathcal{X} \times \{0, 1\}$ and the learner selects $p_t : \mathcal{X} \to [1/T]$. We construct a payoff vector where the first $2T + 2$ components are for the proper calibration objective

$$u_{v,s}(p_t, z_t) = s(y_t - p_t(x_t))\mathrm{Th}_v(p_t(x_t)) \quad \text{for } v \in [1/T], s \in \{+, -\}$$

and the last component captures the sequential multiaccuracy constraint from the sequence of hypothesis functions q_t

$$u_{\mathrm{ma}}(p_t, z_t) = (y_t - p_t(x_t))q_t(x_t)$$

The learner's target set is the set of vectors u whose coordinates are bounded by 0 i.e the negative orthant. We will maintain weights for these $2T + 3$ dimensions of the payoff vector using exponential weights.

Concretely, in Algorithm 2, we use exponential weights update to generate sequence of halfspaces $w^t \in \mathbb{R}^{2T+3}$ with coordinates for every pair $v \in [1/T], s \in \{+, -\}$ and the sequential multiaccuracy constraint. Then for each timestep t, the algorithm constructs the predictor based on the strategy described from Line 12 to Line 18.

The following lemmas are used in the proof of the proposition.

Lemma 3.11 (Halfspace Approachability)**.** For every timestep $t \in [T]$, the predictor $p_t(x)$ described from Line 12 to Line 18 constructed from input q_t satisfies the following guarantee for all $x \in \mathcal{X}, y \in \{0, 1\}$:

$$\mathop{\mathbf{E}}_{p_t}\left[w^t_{\mathrm{sma}} \cdot q_t(x)(y - p_t(x)) + \sum_{v,s} w^t_{v,s} \cdot s \cdot \mathrm{Th}_v(p_t(x)) \cdot (y - p_t(x))\right] \le \frac{1}{T}$$

Proof of Lemma 3.11. Recall the definition of $f(x, j)$ in Line 10. We can rewrite the objective above as showing that for all x, y

$$\mathop{\mathbf{E}}_{j\sim p_t(x)}[f(x, j) \cdot (y - j)] \le \tfrac{1}{T}$$

We now consider the three cases in the strategy:

- Case 1: If $f(x, 0) \le 0$, predict $p_t(x) = 0$: If $f(x, 0) \le 0$, then

$$\begin{aligned}\mathop{\mathbf{E}}_{j\sim p_t}[f(x, j) \cdot (y - j)] &= f(x, 0)y \\ &\le \max\{0, f(x, 0)\} && \text{(for any value of } y \in \{0, 1\}) \\ &\le 0 && \text{(by starting assumption } f(x, 0) \le 0)\end{aligned}$$

- Case 2: If $f(x, 1) > 0$, predict $p_t(x) = 1$: If $f(x, 1) > 0$, then

$$\begin{aligned}\mathop{\mathbf{E}}_{j\sim p_t}[f(x, j) \cdot (y - j)] &= f(x, 1)(y - 1) \\ &\le \max\{0, -f(x, 1)\} && \text{(for any value of } y \in \{0, 1\}) \\ &\le 0 && \text{(by starting assumption } f(x, 1) > 0)\end{aligned}$$

- Case 3: Find adjacent probabilities $\frac{i}{T}, \frac{i+1}{T}$ such that $f\left(x, \frac{i}{T}\right) \cdot f\left(x, \frac{i+1}{T}\right) \leq 0$:

$$
\begin{aligned}
&\mathop{\mathbf{E}}_{j \sim p_t}\left[f(x, j) \cdot (y - j)\right] \\
&= \frac{|f\left(x, \frac{i+1}{T}\right)|}{|f\left(x, \frac{i}{T}\right)| + |f\left(x, \frac{i+1}{T}\right)|} \cdot f(x, \tfrac{i}{T})(y - \tfrac{i}{T}) + \frac{|f(x, \frac{i}{T})|}{|f\left(x, \frac{i}{T}\right)| + |f(x_t, \frac{i+1}{T})|} \cdot f(x, \tfrac{i+1}{T})(y - \tfrac{i+1}{T}) \\
&\leq \frac{|f(x, \frac{i}{T})||f\left(x, \frac{i+1}{T}\right)|}{|f\left(x, \frac{i}{T}\right)| + |f\left(x, \frac{i+1}{T}\right)|} \cdot (1/T) \\
&\leq 1/T \qquad \text{(since } |f(x, j)| \text{ is bounded by 1 for all } j \in [0, 1]\text{)}
\end{aligned}
$$

□

Lemma 3.12 (Halfspace Concentration)**.** The following inequality holds with probability at least $1 - \delta/2$ for the sequence of inputs $(x_1, y_1, q_1), \ldots, (x_T, y_T, q_T)$, weights $w^1, \ldots, w^T$ and outputs $p_1, \ldots, p_T$:

$$
\sum_{t=1}^{T} \left[w^t_{\text{sma}} \cdot q_t(x_t)(y_t - p_t(x_t)) + \sum_{v,s} w^t_{v,s} \cdot s \cdot \text{Th}_v(p_t(x_t)) \cdot (y_t - p_t(x_t)) \right] \leq O\left(\sqrt{T \ln 1/\delta}\right)
$$

Proof of Lemma 3.12. The conclusion of the lemma can be rewritten as showing that

$$
\sum_{t=1}^{T} f(x_t, p_t(x_t))(y_t - p_t(x_t)) \leq O\left(\sqrt{T \ln 1/\delta}\right)
$$

We showed in Lemma 3.11 that $\mathbf{E}_{j \sim p_t(x)}\left[f(x, j) \cdot (y - j)\right] \leq \frac{1}{T}$. Now we apply Azuma-Hoeffding by defining the martingale difference sequence $Z_t = f(x_t, p_t(x_t))(y_t - p_t(x_t)) - \mathbf{E}_{p_t(x_t)}\left[f(x_t, p_t(x_t)) \cdot (y_t - p_t(x_t))\right]$, where the expectation is over the randomness of the sampling of p_t. Note that Z_t is bounded by 2. Applying Azuma-Hoeffding, we have that with probability at least $1 - \delta$,

$$
\sum_{t=1}^{T} Z_t \leq O\left(\sqrt{T \ln 1/\delta}\right)
$$

Since $\sum_{t=1}^{T} \mathbf{E}_{p_t(x_t)}\left[f(x_t, p_t(x_t)) \cdot (y_t - p_t(x_t))\right] \leq T \cdot 1/T = 1$, we obtain the desired result. □

We conclude this subsection with the proof of Proposition 3.10.

Proof of Proposition 3.10. First, we upper bound the expected error, in the sequential setting, of the predictors the algorithm outputs. Let $u(p_t, z_t)$ denote the $2T + 3$ dimensional payoff vector where the first $2T + 2$ coordinates correspond to the proper calibration constraints and the $2T + 3$-th coordinate corresponds to the sequential multiaccuracy constraint. The algorithm maintains a sequence of weight vectors w^t over these $2T + 3$ constraints that ensures the following inequalities are satisfied by any weight vector

$w \geq \mathbf{0}$ with $\|w\|_1 = 1$.

$$
\begin{aligned}
&\left\langle w, \sum_{t\in[T]} u(p_t, z_t)\right\rangle \\
&\leq \sum_{t=1}^{T} \langle w^t, u(p_t, z_t)\rangle + O\left(\sqrt{T \ln T}\right) && \text{(by exponential weights guarantee)} \\
&= \sum_{t=1}^{T}\left[w^t_{\text{sma}} \cdot q_t(x_t)(y_t - p_t(x_t)) + \sum_{v,s} w^t_{v,s} \cdot s \cdot \text{Th}_v(p_t(x_t)) \cdot (y - p_t(x_t))\right] + O\left(\sqrt{T \ln T}\right) \\
&\leq O\left(\sqrt{T \ln 1/\delta}\right) + O\left(\sqrt{T \ln T}\right) && \text{(by Lemma 3.12)} \\
&\leq O\left(\sqrt{T \ln T/\delta}\right)
\end{aligned}
$$

Instantiating the inequality while setting w equal to the first $2T + 2$ standard basis vectors, we obtain an upper bound on the $\mathcal{W}^{1/T}_{\text{Th}}$-weighted calibration error of the sequence $\mathbf{p}, \mathbf{x}, \mathbf{y}$. Observing that each predictor p_t takes values in the set $[1/T] = \{0, \frac{1}{T}, \frac{2}{T}, \ldots, 1\}$ and that the proper calibration error of a $[1/T]$-valued predictor is bounded by twice its $\mathcal{W}^{1/T}_{\text{Th}}$-weighted calibration error Lemma 3.6, we obtain the stated upper bound on $\text{PCalErr}(\mathbf{p}, \mathbf{x}, \mathbf{y})$. Instantiating the inequality while setting w equal to the last standard basis vector, we obtain the upper bound on $\sum_{t=1}^{T} q_t(x_t)(y_t - p_t(x_t))$.

To bound the expected error of $\hat{\mathbf{p}} = \text{unif}(p_1, \ldots, p_T)$ in the distributional setting we again use a martingale argument. Define

$$
Z_t(v, s) = \mathop{\mathbf{E}}_{(x,y)\sim\mathcal{D}}[s \cdot \text{Th}_v(p_t(x))(y - p_t(x))] - s \cdot \text{Th}_v(p_t(x_t))(y_t - p_t(x_t))
$$

for $v \in [1/T], s \in \{+, -\}$. Observe that $Z_t(v, s)$ for $t = 1, \ldots, T$ forms a martingale difference sequence with bounded variance of 2. Similarly, let

$$
Z^{\text{ma}}_t = \mathop{\mathbf{E}}_{(x,y)\sim\mathcal{D}}[q_t(x)(y - p_t(x))] - q_t(x_t)(y_t - p_t(x_t)).
$$

Z^{ma}_t for $t = 1, \ldots, T$ forms a martingale difference sequence with bounded variance of 2. Thus, applying Azuma-Hoeffding's inequality together with union bound over values of $v \in [1/T]$, $s \in \{+, -\}$ and Z^{ma}_t, we get that, with probability at least $1 - \delta$, for all $v \in [1/T]$, $s \in \{+, -\}$

$$
\left|\frac{1}{T}\sum_{t=1}^{T} Z_t(v, s)\right| \leq \sqrt{\frac{\ln \frac{2T}{\delta}}{T}} \qquad \text{and} \qquad \left|\frac{1}{T}\sum_{t=1}^{T} Z^{\text{ma}}_t\right| \leq \sqrt{\frac{\ln \frac{2T}{\delta}}{T}}
$$

The distributional error bounds now follow by observing that

$$
\begin{aligned}
\forall v, s \quad \frac{1}{T}\sum_{t=1}^{T} \mathop{\mathbf{E}}_{(x,y)\sim\mathcal{D}}[s \cdot \text{Th}_v(p_t(x))(y - p_t(x))] &= \frac{1}{T}\sum_{t=1}^{T} s \cdot \text{Th}_v(p_t(x_t))(y_t - p_t(x_t)) \;+\; \frac{1}{T}\sum_{t=1}^{T} Z_t(v, s) \\
\frac{1}{T}\sum_{t=1}^{T} \mathop{\mathbf{E}}_{(x,y)\sim\mathcal{D}}[q_t(x)(y - p_t(x)] &= \frac{1}{T}\sum_{t=1}^{T} q_t(x_t)(y_t - p_t(x_t)) \;+\; \frac{1}{T}\sum_{t=1}^{T} Z^{\text{ma}}_t
\end{aligned}
$$

and that we have already shown all terms on the right side of both inequalities are bounded above by $O\left(\sqrt{\frac{\log T/\delta}{T}}\right)$.

The algorithm runs in poly(T) timesteps per iteration and each p_t is simply a remapping of q_t i.e $p_t(x) = u(q_t(x))$ for some (random) function $u : [-1, 1] \to [1/T]$. To see this observe that in the algorithm $p_t(x)$ depends on x via $q_t(x)$ (see definition of $f(x, j)$). Additionally, the function u can be computed in polynomial time using the weights w^t by implementing the logic in Lines 10–17. □

4 Online Omniprediction

In this section, we show that online omniprediction for a class of hypothesis functions $\mathcal{H}$ and a class of loss function $\mathcal{L}$ can be achieved using an online weak agnostic learner for $\Delta\mathcal{L} \circ \mathcal{H}$ with regret that scales with $\tilde{O}\left(\sqrt{T}\right)$ and the regret of the online weak agnostic learner. Online weak agnostic learning, as studied in (Chen et al., 2012; Brukhim et al., 2020; Beygelzimer et al., 2015), outputs a sequence of predictions whose correlation with the outcome sequence is not much less than that of the best fixed hypothesis in hindsight.

Definition 4.1 (Online Weak Agnostic Learner)**.** Given a hypothesis class $C \subset \{c : \mathcal{X} \to [-1, 1]\}$, a failure probability δ, and an *online* sequence of context-label pairs $(x_1, y_1), \ldots, (x_T, y_T)$ where $x_t \in \mathcal{X}$ and $y_t : [-1, 1]$, an online weak agnostic learner outputs a sequence of predictors $q_1, \ldots, q_T : \mathcal{X} \to [-1, 1]$ such that with probability at least $1 - \delta$ over the randomness of the learner and a possibly adaptive adversary, the following holds:

$$\max_{c \in C} \sum_{t=1}^{T} c(x_t)y_t \leq \sum_{t=1}^{T} q_t(x_t)y_t + \text{OracleReg}_C^{\delta}(T)$$

Using an online weak agnostic learner for $\Delta\mathcal{L} \circ \mathcal{H}$, we present the following algorithm for omniprediction.

Algorithm 3 Online Omniprediction

procedure OnlineOmniprediction(T)
  Input: Sequence of samples $(x_1, y_1), \ldots, (x_T, y_T)$
  Input: A failure probability parameter, $\delta > 0$.
  Input: An online weak agnostic learner, $\mathcal{A}_{\text{owal}}$, for function class $C = \{-1, +1\} \cdot \Delta\mathcal{L} \circ \mathcal{H}$ with failure probability $\delta/2$
  Input: An augmented proper calibrator (Algorithm 2), $\mathcal{A}_{\text{apcal}}$, with failure probability $\delta/2$
  Output: Sequence of predictors $p_1, \ldots, p_T : \mathcal{X} \to [0, 1]$
  Receive q_1 from $\mathcal{A}_{\text{owal}}$
  for $t = 1$ to T **do**
    Send q_t and sample (x_t, y_t) to $\mathcal{A}_{\text{apcal}}$
    Receive p_t from $\mathcal{A}_{\text{apcal}}$
    Send sample $(x_t, y_t - p_t(x_t))$ to $\mathcal{A}_{\text{owal}}$
    Receive q_{t+1} from $\mathcal{A}_{\text{owal}}$

Theorem 4.1. Given an online sequence of samples $(x_1, y_1), \ldots, (x_T, y_T)$, a class of loss functions $\mathcal{L}$, a class of hypothesis functions $\mathcal{H}$, an online weak agnostic learner for $\Delta\mathcal{L} \circ \mathcal{H}$, and a failure probability δ, Algorithm 3 can be initialized to guarantee $(\mathcal{L}, \mathcal{H})$−online omniprediction regret with probability at least $1 - \delta$

$$O\left(\sqrt{T \ln T/\delta}\right) + \text{OracleReg}_C^{\delta}(T)$$

Proof of Theorem 4.1. Let $C = \Delta\mathcal{L} \circ \mathcal{H}$. The online weak agnostic learner interacting with the online sequence guarantees with probability at least $1 - \delta$

$$\max_{c \in C} \sum_{t=1}^{T} c(x_t)(y_t - p_t(x_t)) \leq \sum_{t=1}^{T} q_t(x_t)(y_t - p_t(x_t)) + \text{OracleReg}_C^{\delta}(T)$$

since Algorithm 3 sets the labeling function to $r_t(x, y) = y - p_t(x)$. The Augmented Proper Calibration algorithm guarantees with probability at least $1 - \delta$, $\text{PCalErr}(\mathbf{p}, \mathbf{x}, \mathbf{y}) \leq O\left(\sqrt{T \ln T/\delta}\right)$ and $\sum_{t=1}^{T} q_t(x_t)(y_t - p_t(x_t)) \leq O\left(\sqrt{T \ln T/\delta}\right)$ as stated in Proposition 3.10. The second guarantee is the same as the first term RHS on the online weak agnostic learner guarantee. Thus, we obtain

$$\max_{c \in C} \sum_{t=1}^{T} c(x_t)(y_t - p_t(x_t)) \leq O\left(\sqrt{T \ln 1/\delta}\right) + \text{OracleReg}_C^{\delta}(T)$$

This allows us to conclude proper calibration and C-multiaccuracy, which together implies $(\mathcal{L}, \mathcal{H})$-OmniRegret $\leq O\left(\sqrt{T \ln 1/\delta}\right) + \text{OracleReg}_C^{\delta}(T)$ □

Corollary 4.2. Given a hypothesis class $\mathcal{H}$ and a class of loss functions $\mathcal{L}$ such that the composed class $\Delta\mathcal{L} \circ \mathcal{H}$ has bounded sequential Rademacher complexity, then there exists a forecaster that guarantees expected omniprediction regret bounded by

$$O\left(\sqrt{T \ln T} + \mathsf{srad}_T(\Delta\mathcal{L} \circ \mathcal{H})\right)$$

where $\mathsf{srad}_T(C)$ refers to the sequential Rademacher complexity of a class C.

4.1 Online Weak Agnostic Learning

In this subsection, we discuss the online weak agnostic learning problem in Definition 4.1, showing the existence of online weak agnostic learners whose regret scales optimally with the sequential rademacher complexity of the hypothesis class.

First, we note that when the labels are boolean valued and the hypothesis class are constrained to be boolean functions, online weak agnostic learning is equivalent to agnostic online binary classification problem where in each iteration the learner suffers loss of $\mathbf{1}[\hat{y}_t \neq y_t]$ and the goal is to minimize the cummulate loss.

When C is finite, a multiplicative weights algorithm can be used to achieve online weak agnostic learning with regret bounded by $O\left(\sqrt{T \ln |C|}\right)$.

Lemma 4.3 (Multiplicative Weights)**.** Given a finite hypothesis class C, Algorithm 4 implements an online weak agnostic learner whose regret scales as $O\left(\sqrt{T \ln |C|}\right)$ with probability 1.

Algorithm 4 Multiplicative Weights for Online Weak Agnostic Learning

Input: Finite hypothesis class C, learning rate $\eta = \sqrt{\frac{\ln |C|}{T}}$
Initialize weights $w_1(c) = \frac{1}{|C|}$ for all $c \in C$
for $t = 1, \ldots, T$ **do**
    Receive context x_t
    Output prediction $q_t(x_t) = \sum_{c \in C} w_t(c) c(x_t)$
    Observe label $y_t \in [-1, 1]$
    Update weights: $w_{t+1}(c) = \frac{w_t(c) \exp(\eta c(x_t) y_t)}{\sum_{c' \in C} w_t(c') \exp(\eta c'(x_t) y_t)}$

For infinite hypothesis classes, the achievable regret values is characterized by the sequential Rademacher complexity of the hypothesis class.

Theorem 4.4. Given a hypothesis class C with finite sequential Rademacher complexity, there exists an algorithm that implements an online weak agnostic learner with regret that scales as $O(T \cdot \mathsf{srad}_T(C))$. Moreover, no algorithm can do better than $\Omega(T \cdot \mathsf{srad}_T(C))$.

To prove this theorem, we need to formulate the rates for online weak agnostic learning as the minimax value of a game.

Definition 4.2 (Minimax Value of Online Weak Agnostic Learning)**.** Given a hypothesis class $C \subset \{c : \mathcal{X} \to [-1, 1]\}$. Consider randomized learners who predict a distribution $q_t \in \Delta(\mathcal{X} \to [-1, 1])$ and sample p_t from this distribution on every round t. We define the value of the game as

$$\mathcal{V}_T^{\text{owal}}(C) = \inf_{p_1 \in Q} \sup_{(x_1, y_1) \in \mathcal{X} \times \mathcal{Y}} \mathbb{E}_{p_1 \sim q_1} \cdots \inf_{q_T \in Q} \sup_{(x_T, y_T) \in \mathcal{X}, \mathcal{Y}} \mathbb{E}_{p_T \sim q_T} \left[\sup_{c \in C} \sum_{t=1}^{T} c(x_t) y_t - \sum_{t=1}^{T} p_t(x_t) y_t \right] \tag{1}$$

Proof of Theorem 4.4. Since the losses are linear and 1-Lipschitz in the variable y_t, the upper bound result follows by applying Theorem 8 of Rakhlin et al. (2015) to bound the sequential Rademacher complexity of the hypothesis class. Thus,

$$\mathcal{V}_T^{\text{owal}}(C) \leq 2T \cdot \mathsf{srad}_T(C)$$

To show the lower bound we simply choose the label sequence $y_1, \ldots, y_T$ to be i.i.d. Rademacher random variables, thus, it immediately follows that for any sequence $x_1, \ldots, x_T$

$$\mathcal{V}_T^{\text{owal}}(C) \geq \mathop{\mathbf{E}}_{y_1, \ldots, y_T} \left[\sup_{c \in C} \sum_{t=1}^{T} c(x_t) y_t - \sum_{t=1}^{T} p_t(x_t) y_t \right]$$

Since $y_1, \ldots, y_T$ are i.i.d. Rademacher random variables, the second term becomes zero in expectation regardless of the choice of p_t sequence, giving us the desired result.

$$\mathcal{V}_T^{\text{owal}}(C) \geq T \cdot \mathsf{srad}_T(C)$$

□

Remark 4.5. For a finite class C the unnormalized sequential Rademacher complexity satisfies $T \cdot \mathsf{srad}_T(C) \leq O\left(\sqrt{T \log |C|}\right)$, hence Theorem 4.4 recovers the same regret bound as Lemma 4.3, albeit without supplying an explicit algorithm.

Due to the linearity of the loss function and the fact that the online weak agnostic learner is allowed to output values in the $[-1, 1]$ interval, there exist *deterministic* online weak agnostic learners for any class C that is online learnable. This is because we can always predict the expectation of the *randomized* online weak agnostic learner and achieve the same guarantee. However, such deterministic online weak agnostic learner might be more computationally difficult to implement, so we will only assume access to a *randomized* online weak agnostic learner.

Online Weak Agnostic Learning for $\{-1, +1\} \times C$. Our algorithms for omniprediction will actually use an online weak agnostic learning oracle for $\{-1, +1\} \times C$. However, one should think of this as equivalent to C, because of the following lemma.

Lemma 4.6. Given an online weak agnostic learner for C, one can construct an online weak agnostic learner for $\{-1, +1\} \times C$ that uses two calls to the original learner for C at each round. The resulting algorithm's regret satisfies

$$\mathrm{OracleReg}^{\delta}_{\{-1,+1\}\times C}(T) \;\le\; \mathrm{OracleReg}^{\delta}_{C}(T) \;+\; O(\sqrt{T}).$$

Proof. The main idea is as follows: we run two copies of the online weak agnostic learner for C, one with the labels unchanged ("positive sign") and one with the labels negated ("negative sign"), and then apply multiplicative weights to combine their predictions.

Concretely, let $\textsc{owal}^{+}$ be the online weak agnostic learner run on the sequence $\{(x_1, y_1), \ldots, (x_{t-1}, y_{t-1})\}$, and let $\textsc{owal}^{-}$ be another instance run on $\{(x_1, -y_1), \ldots, (x_{t-1}, -y_{t-1})\}$. In each round t:

1. We query $\textsc{owal}^{+}$ to get a predictor q_t^{+} and query $\textsc{owal}^{-}$ to get a predictor q_t^{-}.
2. We maintain an exponential-weight "score" for each of the two experts, reflecting their cumulative performance so far. Specifically, we use a learning rate $\eta = T^{-1/2}$, and multiply the weight of the positive-sign expert by $\exp(\eta \sum_{u=1}^{t-1} q_u^{+}(x_u)\, y_u)$, whereas the weight of the negative-sign expert is multiplied by $\exp(-\eta \sum_{u=1}^{t-1} q_u^{-}(x_u)\, y_u)$.
3. We normalize these two weights and form a convex combination:

$$q_t \;=\; \alpha_t\, q_t^{+} \;+\; (1 - \alpha_t)\, q_t^{-},$$

   where

$$\alpha_t \;=\; \frac{\exp(\eta \sum_{u=1}^{t-1} q_u^{+}(x_u)\, y_u)}{\exp(\eta \sum_{u=1}^{t-1} q_u^{+}(x_u)\, y_u) \;+\; \exp(-\eta \sum_{u=1}^{t-1} q_u^{-}(x_u)\, y_u)}.$$

4. We output the single deterministic predictior q_t for round t, and after seeing y_t, update the two experts' weights accordingly.

By the usual analysis of multiplicative weights, the combination's cumulative reward $\sum_{t=1}^{T} q_t\, y_t$ is within $O(\sqrt{T})$ of the better of the two experts (i.e., $\textsc{owal}^{+}$ or $\textsc{owal}^{-}$). Moreover, each of these two experts has online weak agnostic regret at most $\mathrm{OracleReg}^{\delta}_{C}(T)$ relative to the best function in C with sign fixed. Thus, we obtain

$$\sum_{t=1}^{T} q_t(x_t)\, y_t \;\;\ge\;\; \max_{\substack{s\in\{+,-\}\\ c\in C}} \sum_{t=1}^{T} s\, c(x_t)\, y_t \;-\; \mathrm{OracleReg}^{\delta}_{C}(T) \;-\; O(\sqrt{T}).$$

Rearranging yields the stated bound on $\mathrm{OracleReg}^{\delta}_{\{-1,+1\}\times C}(T)$. □

4.2 Omniprediction for Finite Classes

As an easy first application of Theorem 4.1, we assume the loss class $\mathcal{L}$ and hypothesis class $\mathcal{H}$ are both finite, and we obtain online omnipredictors whose regret depends near-optimally on T and only logarithmically on $|\mathcal{L}|$ and $|\mathcal{H}|$.

Corollary 4.7 (Omniprediction for Finite Loss Classes)**.** Given a finite class of hypothesis functions $\mathcal{H}$ and a finite class of bounded loss functions $\mathcal{L}$, Algorithm 3 guarantees expected omniprediction regret of $O\left(\sqrt{T \ln(|\mathcal{H}| \cdot |\mathcal{L}| \cdot T)}\right)$ using Algorithm 4 as the online weak agnostic learning oracle.

The proof of the corollary is immediate by combining Theorem 4.1 with Lemma 4.3 and observing that the cardinality of the class $\Delta\mathcal{L} \circ \mathcal{H}$ is bounded above by $|\mathcal{H}| \cdot |\mathcal{L}|$.

4.3 Omniprediction for Infinite Loss Classes with Finite Approximate Basis

While achieving omniprediction for infinite loss classes might seem challenging, a key observation is that if a class of loss functions $\mathcal{F}$ can be approximated by a simpler set of "basis functions" $\mathcal{G}$, then multiaccuracy for $\Delta\mathcal{G} \circ \mathcal{H}$ implies multiaccuracy for $\Delta\mathcal{F} \circ \mathcal{H}$ with bounded excess regret. In the sequel, we will apply this abstract approximation result, taking our basis functions to be thresholds or ReLUs (or finite subsets thereof) and using them to approximate important infinite classes of losses.

Definition 4.3 (Approximate Basis)**.** Let Γ be a set and $\mathcal{F} = \{f : \Gamma \to [-1, 1]\}$ a class of functions on Γ. We say that a set $\mathcal{G} = \{g : \Gamma \to [-1, 1]\}$ is an *ε-basis for $\mathcal{F}$ with sparsity d and coefficient norm λ*, if for every function $f \in \mathcal{F}$, there exists a finite subset $\{g_1, \ldots, g_d\} \subseteq \mathcal{G}$ and coefficients $c_1, c_2, \ldots, c_d \in [-1, 1]$ satisfying

$$\forall x \in \Gamma \quad \left| f(x) - \sum_{i=1}^{d} c_i g_i(x) \right| \leq \varepsilon \qquad \text{and} \qquad \sum_{i=1}^{d} |c_i| \leq \lambda. \tag{31}$$

In the special case when $\mathcal{G}$ itself has d elements, we say $\mathcal{G}$ is a *finite ε-basis for $\mathcal{F}$ of **size** d and coefficient norm λ*.

A useful property of approximate bases is that the approximation is preserved under post-composition with any class of functions.

Lemma 4.8 (Finite approximate bases are preserved under post-composition)**.** Let Γ_0, Γ_1 be sets and $\mathcal{H} = \{h : \Gamma_0 \to \Gamma_1\}$ a class of functions from Γ_0 to Γ_1. If $\mathcal{F} = \{f : \Gamma_1 \to [-1, 1]\}$ is any class of functions on Γ_1 and $\mathcal{G} = \{g : \Gamma_1 \to [-1, 1]\}$ is an ε-basis for $\mathcal{F}$ with sparsity d and coefficient norm λ, then $\mathcal{G} \circ \mathcal{H}$ is an ε-basis for $\mathcal{F} \circ \mathcal{H}$ with sparsity d and coefficient norm λ.

Proof. Consider any $f \in \mathcal{F}$. If coefficients $c_1, \ldots, c_d$ satisfy property (31) in the definition of ε-basis with coefficient norm λ, then for all $h \in \mathcal{H}$, we have

$$\forall x \in \Gamma_0 \quad \left| f(h(x)) - \sum_{i=1}^{d} c_i g_i(h(x)) \right| \leq \varepsilon \qquad \text{and} \qquad \sum_{i=1}^{d} |c_i| \leq \lambda.$$

which confirms that $\mathcal{G} \circ \mathcal{H}$ is an ε-basis for $\mathcal{F} \circ \mathcal{H}$ with coefficient norm 1. □

Approximate bases allow us to extend multiaccuracy from the basis functions to the entire class $\mathcal{F}$.

Lemma 4.9. Let $\mathcal{F} = \{f : [0,1] \to [-1,1]\}$ be a class of functions and $\mathcal{G} = \{g_1, \ldots, g_d : [0,1] \to [-1,1]\}$ an ε-basis for $\mathcal{F}$ with sparsity d and coefficient norm λ. Then for any sequence of T predictions $\mathbf{p}$ and context, outcome pairs $\mathbf{x}, \mathbf{y}$

$$\mathcal{F}\text{-MAErr}(\mathbf{p}, \mathbf{x}, \mathbf{y}) \le \lambda \cdot \mathcal{G}\text{-MAErr}(\mathbf{p}, \mathbf{x}, \mathbf{y}) + \varepsilon T$$

Similarly, in the distributional setting, for any distribution $\mathcal{D}$ on $\mathcal{X} \times [0,1]$ and any predictor $p : \mathcal{X} \to [0,1]$,

$$\mathcal{F}\text{-MAErr}_{\mathcal{D}}(p) \le \lambda \cdot \mathcal{G}\text{-MAErr}_{\mathcal{D}}(p) + \varepsilon.$$

Proof. We can write every $f \in \mathcal{F}$ as follows: $f(z) = \sum_{i \in [d]} c_i(f) g_i(z) + \varepsilon(f)$ such that $\sum_{i \in [d]} |c_i(f)| \le \lambda$, $|\varepsilon(f)| \le \varepsilon$ and $g_1, \ldots g_d \in \mathcal{G}$. Consequently,

$$\begin{aligned}
\max_{f \in \mathcal{F}} \left| \sum_{t=1}^{T} f(x_t)(y_t - p_t(x_t)) \right| &= \max_{f \in \mathcal{F}} \left| \sum_{t=1}^{T} \left(\sum_{i \in [d]} c_i(f) g_i(x_t)(y_t - p_t(x_t)) \right) + \varepsilon(f)(y_t - p_t(x_t)) \right| && (32) \\
&\le \max_{f \in \mathcal{F}} \left| \sum_{i \in [d]} c_i(f) \left(\sum_{t=1}^{T} g_i(x_t)(y_t - p_t(x_t)) \right) \right| + \left| \sum_{t=1}^{T} \varepsilon(f)(y_t - p_t(x_t)) \right| && (33) \\
&\le \max_{f \in \mathcal{F}} \sum_{i \in [d]} |c_i(f)| \left| \sum_{t=1}^{T} g_i(x_t)(y_t - p_t(x_t)) \right| + \varepsilon T && (\text{since } |\varepsilon(f)| \le \varepsilon) \\
&\le \max_{f \in \mathcal{F}} \left(\sum_{i \in [d]} |c_i(f)| \right) \max_{i \in [d]} \left| \sum_{t=1}^{T} g_i(x_t)(y_t - p_t(x_t)) \right| + \varepsilon T && (34) \\
&= \lambda \max_{g \in \mathcal{F}_{\text{basis}}} \left| \sum_{t=1}^{T} g(x_t)(y_t - p_t(x_t)) \right| + \varepsilon T && (35)
\end{aligned}$$

The bound for multiaccuracy error in the distributional setting follows by an identical calculation, substituting expectations over $\mathcal{D}$ for sums over $t \in \{1, \ldots, T\}$. □

To illustrate the application of Lemma 4.9, we note the following consequence for $(\mathcal{L}, \mathcal{H})$-online omniprediction when $|\mathcal{H}| < \infty$ and $\Delta\mathcal{L}$ has a finite approximate basis.

Proposition 4.10 (Omniprediction for Infinite Loss Classes and Finite Hypothesis Classes)**.** Let $\mathcal{H}$ be a finite class of hypothesis functions. Let $\mathcal{L}$ be a (possibly infinite) class of loss functions whose discrete derivative class $\Delta\mathcal{L}$ admits a finite ε-basis of size d with coefficient norm λ. There exists a forecaster that guarantees $(\mathcal{L}, \mathcal{H})$-OmniRegret $\le O\left(\lambda \sqrt{T \ln(|\mathcal{H}| \cdot dT)} + \varepsilon T\right)$.

Proof of Proposition 4.10. Let $\mathcal{G}$ denote the finite ε-basis of $\Delta\mathcal{L}$ of size d and coefficient norm λ. By Lemma 4.8, $\mathcal{G} \circ \mathcal{H}$ is an ε-basis of $\Delta\mathcal{L} \circ \mathcal{H}$ with the same sparsity and coefficient norm. We run Algorithm 3 with hypothesis class $\mathcal{G} \circ \mathcal{H}$ instead of $\Delta\mathcal{L} \circ \mathcal{H}$.

By Corollary 4.7, this algorithm guarantees expected proper calibration error and expected $(\mathcal{G} \circ \mathcal{H})$-multiaccuracy error of $O\left(\sqrt{T \ln(|\mathcal{H}| \cdot dT)}\right)$.

Now, we use Lemma 4.9 to bound the $(\Delta\mathcal{L}\circ\mathcal{H})$-multiaccuracy error in terms of the $(\mathcal{G}\circ\mathcal{H})$-multiaccuracy error.

$$(\Delta\mathcal{L}\circ\mathcal{H})\text{-MAErr} \le \lambda(\mathcal{G}\circ\mathcal{H})\text{-MAErr}(\mathbf{p},\mathbf{x},\mathbf{y}) + \varepsilon T$$

Combining these bounds gives the desired bound of $O\left(\lambda\sqrt{T\ln(|\mathcal{H}|\cdot dT)} + \varepsilon T\right)$ on the $(\Delta\mathcal{L}\circ\mathcal{H})$-multiaccuracy error of the forecaster. □

Proposition 4.11 (Omniprediction for Infinite Loss and Hypothesis Classes)**.** For any loss class $\mathcal{L}$ and hypothesis class $\mathcal{H}$, if we are given an online weak agnostic learning oracle for $\mathcal{G}\circ\mathcal{H}$ where $\mathcal{G}$ is an ε-basis of $\Delta\mathcal{L}\circ\mathcal{H}$ with coefficient norm λ, then the application of Algorithm 3 with hypothesis class $\mathcal{G}\circ\mathcal{H}$ yields an omnipredictor with $(\mathcal{L},\mathcal{H})$-OmniRegret $\le O\left(\sqrt{T\ln T} + \varepsilon T + \lambda\cdot\text{OracleReg}_{\mathcal{G}\circ\mathcal{H}}(T)\right)$.

Proof. The regret bound is a direct application of Theorem 4.1 combined with Lemma 4.9. □

In Section 6.1, we will need to make use of a generalization of Definition 4.3 that allows approximating functions in $\mathcal{F}$ using *infinite* linear combinations of elements of the function class $\mathcal{G}$. For future reference, we provide the generalization here.

Definition 4.4. Let Γ be a set and $\mathcal{F} = \{f : \Gamma \to [-1,1]\}$ a class of functions on Γ. Let (Ω,Σ) be a measurable space and $\mathcal{G} = \{g_\omega : \Gamma \to [-1,1] \mid \omega\in\Omega\}$ a class of functions on Γ indexed by Ω. We say $\mathcal{F}$ is $(\mathcal{G},\varepsilon)$-spanned with coefficient norm λ if for every $f\in\mathcal{F}$ there is a signed measure c_f on (Ω,Σ) such that $\|c_f\| = |c_f|(\Omega) \le \lambda$ and

$$\forall x\in\Gamma \quad \left|f(x) - \int_\Omega g_\omega(x)\,\mathrm{d}c_f(\omega)\right| \le \varepsilon.$$

Lemma 4.9 generalizes to $(\mathcal{G},\varepsilon)$-spanned classes under this definition, as follows.

Lemma 4.12. Let (Ω,Σ) be a measurable space and and $\mathcal{G} = \{g_\omega : \Gamma\to[0,1] \mid \omega\in\Omega\}$ a class of functions indexed by Ω. Suppose $\mathcal{F} = \{f : [0,1]\to[-1,1]\}$ is a class of functions that is $(\mathcal{G},\varepsilon)$-spanned with coefficient norm λ. Then for any sequence of T predictions $\mathbf{p}$ and context, outcome pairs $\mathbf{x},\mathbf{y}$

$$\mathcal{F}\text{-MAErr}(\mathbf{p},\mathbf{x},\mathbf{y}) \le \lambda\cdot\mathcal{G}\text{-MAErr}(\mathbf{p},\mathbf{x},\mathbf{y}) + \varepsilon T$$

Similarly, in the distributional setting, for any distribution $\mathcal{D}$ on $\mathcal{X}\times[0,1]$ and any predictor $p:\mathcal{X}\to[0,1]$,

$$\mathcal{F}\text{-MAErr}_{\mathcal{D}}(p) \le \lambda\cdot\mathcal{G}\text{-MAErr}_{\mathcal{D}}(p) + \varepsilon.$$

Proof. We can write every $f\in\mathcal{F}$ as follows: $f(z) = \int_\Omega g_\omega(z)\,\mathrm{d}c_f(\omega) + \varepsilon_f(z)$ such that $|c|(\Omega)\le\lambda$, $\|\varepsilon_f\|_\infty \le \varepsilon$

and $g_\omega \in \mathcal{G}$ for all $\omega \in \Omega$. Consequently,

$$\sup_{f\in\mathcal{F}}\left|\sum_{t=1}^{T} f(x_t)(y_t - p_t(x_t))\right| = \sup_{f\in\mathcal{F}}\left|\sum_{t=1}^{T}\left(\int_\Omega g_\omega(x_t)(y_t - p_t(x_t))\,\mathrm{d}c_f(\omega)\right) + \varepsilon_f(x_t)(y_t - p_t(x_t))\right| \tag{36}$$

$$\leq \sup_{f\in\mathcal{F}}\left\{\left|\int_\Omega\left(\sum_{t=1}^{T} g_\omega(x_t)(y_t - p_t(x_t))\right)\mathrm{d}c_f(\omega)\right| + \left|\sum_{t=1}^{T}\varepsilon_f(x_t)(y_t - p_t(x_t))\right|\right\} \tag{37}$$

$$\leq \sup_{f\in\mathcal{F}}\int_\Omega\left|\sum_{t=1}^{T} g_\omega(x_t)(y_t - p_t(x_t))\right|\mathrm{d}c_f(\omega) \;+\; \varepsilon T$$

$$\text{(since } |\varepsilon_f(x_t)| \leq \varepsilon,\ |y_t - p_t(x_t)| \leq 1\text{)}$$

$$\leq \sup_{f\in\mathcal{F}}\|c_f\| \cdot \sup_{\omega\in\Omega}\left|\sum_{t=1}^{T} g_\omega(x_t)(y_t - p_t(x_t))\right| \;+\; \varepsilon T \tag{38}$$

$$\leq \lambda \sup_{g\in\mathcal{G}}\left|\sum_{t=1}^{T} g(x_t)(y_t - p_t(x_t))\right| + \varepsilon T \tag{39}$$

The bound for multiaccuracy error in the distributional setting follows by an identical calculation, substituting expectations over $\mathcal{D}$ for sums over $t \in \{1, \ldots, T\}$.

$$\sup_{f\in\mathcal{F}}\left|\mathop{\mathbf{E}}_{(x,y)\sim\mathcal{D}}[f(x)(y - p(x))]\right| = \sup_{f\in\mathcal{F}}\left|\mathop{\mathbf{E}}_{(x,y)\sim\mathcal{D}}\left[\left(\int_\Omega g_\omega(x)(y - p(x))\,\mathrm{d}c_f(\omega)\right) + \varepsilon_f(x)(y - p(x))\right]\right| \tag{40}$$

$$\leq \sup_{f\in\mathcal{F}}\left\{\left|\int_\Omega \mathop{\mathbf{E}}_{(x,y)\sim\mathcal{D}}[g_\omega(x)(y - p(x))]\;\mathrm{d}c_f(\omega)\right| + \left|\mathop{\mathbf{E}}_{(x,y)\sim\mathcal{D}}\left[\varepsilon_f(x)(y - p(x))\right]\right|\right\} \tag{41}$$

$$\leq \sup_{f\in\mathcal{F}}\int_\Omega\left|\mathop{\mathbf{E}}_{(x,y)\sim\mathcal{D}}[g_\omega(x)(y - p(x))]\right|\mathrm{d}c_f(\omega) \;+\; \varepsilon$$

$$\text{(since } |\varepsilon_f(x)| \leq \varepsilon, |y - p(x)| \leq 1\text{)}$$

$$\leq \sup_{f\in\mathcal{F}}\|c_f\| \cdot \sup_{\omega\in\Omega}\left|\mathop{\mathbf{E}}_{(x,y)\sim\mathcal{D}}[g_\omega(x)(y - p(x))]\right| \;+\; \varepsilon \tag{42}$$

$$\leq \lambda \sup_{g\in\mathcal{G}}\left|\mathop{\mathbf{E}}_{(x,y)\sim\mathcal{D}}[g(x)(y - p(x))]\right| + \varepsilon \tag{43}$$

□

5 Omniprediction for Notable Loss Classes

This section instantiates Proposition 4.10 for concrete classes of losses, by exhibiting approximate bases with appropriate parameters. In particular, we focus on the classes of all 1-Lipschitz convex losses ($\mathcal{L}_{\mathrm{cvx}}$), all 1-Lipschitz losses ($\mathcal{L}_{\mathrm{lip}}$), and all bounded variation losses ($\mathcal{L}_{\mathrm{BV}}$) including proper losses ($\mathcal{L}_{\mathrm{proper}}$). The bases we develop in this section will also be used in developing efficient offline algorithms for omniprediction with respect to these infinite classes.

5.1 Online Omniprediction bounds for Notable Loss Classes

Theorem 5.1. Let $\mathcal{H} = \{h : X \to [0,1]\}$ be a hypothesis class.

1. Given an online weak agnostic learning oracle for $\mathrm{ReLU}^{1/T} \circ \mathcal{H}$, Algorithm 3 implements an online omnipredictor for loss class $\mathcal{L}_{\mathrm{cvx}}$ with regret
$$(\mathcal{L}_{\mathrm{cvx}}, \mathcal{H})\text{-OmniRegret} \le O\left(\sqrt{T \ln T} + \mathrm{OracleReg}_{\mathrm{ReLU} \circ \mathcal{H}}(T)\right). \tag{44}$$
2. Given an online weak agnostic learning oracle for $\mathrm{Th}^{1/T} \circ \mathcal{H}$, Algorithm 3 implements an online omnipredictor for loss class $\mathcal{L}_{\mathrm{lip}}$ with regret
$$(\mathcal{L}_{\mathrm{lip}}, \mathcal{H})\text{-OmniRegret} \le O\left(\sqrt{T \ln T} + \mathrm{OracleReg}_{\mathrm{Th}^{1/T} \circ \mathcal{H}}(T)\right). \tag{45}$$
3. Suppose the hypotheses in $\mathcal{H}$ are Γ-valued, where Γ is a finite subset of $[0,1]$ containing $\{0,1\}$. Given an online weak agnostic learning oracle for $\mathrm{Th}^{\Gamma} \circ \mathcal{H}$, Algorithm 3 implements an online omnipredictor for loss class $\mathcal{L}_{\mathrm{BV}}$ with regret
$$(\mathcal{L}_{\mathrm{BV}}, \mathcal{H})\text{-OmniRegret} \le O\left(\sqrt{T \ln T} + \mathrm{OracleReg}_{\mathrm{Th}^{\Gamma} \circ \mathcal{H}}(T)\right). \tag{46}$$

If $\mathcal{H}$ is a finite hypothesis class, then the omniprediction bounds in parts 1 and 2 above are $O\left(\sqrt{T \ln(|\mathcal{H}| \cdot T)}\right)$, and the bound in part 3 is $O\left(\sqrt{T \ln(|\Gamma| \cdot |\mathcal{H}| \cdot T)}\right)$.

Proof. The bounds (44), (45), (46) follow directly from application of Theorem 4.1 with Corollary 5.3, Lemma 5.4, and Lemma 5.7, respectively. The bounds for finite hypothesis classes follow by applying Proposition 4.10. □

To aid in interpreting the regret bounds (44)-(46), we remind the reader that for any class C there exists a (not necessarily computationally efficient) online weak agnostic learning oracle satisfying $\mathrm{OracleReg}_C(T) = O(T \cdot \mathsf{srad}_T(C))$, where srad denotes sequential Rademacher complexity.

5.2 Approximate Bases for Notable Loss Classes

In this subsection we exhibit approximate bases for the loss classes listed above. The quantitative consequences for omniprediction will be detailed in the following section.

Convex Lipschitz Loss Functions. Let $\mathcal{F}_{\mathrm{cvx}}$ be the class of all convex 1-Lipschitz functions.

Lemma 5.2 (Gopalan et al. (2024b))**.** For all $\varepsilon > 0$, $\mathcal{F}_{\mathrm{cvx}}$ admits a finite ε-basis of ReLU functions of size $\tilde{O}(1/\varepsilon^{2/3})$ with coefficient norm 2.

Convex functions are not closed under linear combinations. This means that not all functions in $\Delta\mathcal{L}_{\mathrm{cvx}}$ will be convex, therefore we cannot apply the result from section above. However, the fact that this class is derived from the difference of two convex functions still allows us to derive useful upper bounds.

Corollary 5.3 (of Lemma 5.2)**.** For all $\varepsilon > 0$, $\Delta\mathcal{L}_{\mathrm{cvx}}$ admits a finite ε-basis of ReLU functions of size $\tilde{O}(1/\varepsilon^{2/3})$ with coefficient norm 4.

Proof. Let $\mathcal{G} \subset \text{ReLU}$ be a finite $\varepsilon/2$-basis for $\mathcal{F}_{\text{cvx}}$ of size $\tilde{O}(1/\varepsilon^{2/3})$ with coefficient norm 2. Denote the elements of $\mathcal{G}$ as $g_1, \ldots, g_d$ where $d = \tilde{O}(1/\varepsilon^{2/3})$. If $f \in \Delta\mathcal{L}_{\text{cvx}}$ then $f(x) = f_1(x) - f_0(x)$ where $f_0, f_1 \in \mathcal{F}_{\text{cvx}}$. Let c_{0i}, c_{1i} $(1 \le i \le d)$ be coefficients such that $\|f_0 - \sum_{i=1}^d c_{0i} g_i\|_\infty \le \varepsilon/2$, $\|f_1 - \sum_{i=1}^d c_{1i} g_i\|_\infty \le \varepsilon/2$, $\sum_{i=1}^d |c_{0i}| \le 2$, and $\sum_{i=1}^d |c_{1i}| \le 2$. Then for the coefficients $c_i = c_{1i} - c_{0i}$ we have

$$\left\| f - \sum_{i=1}^d c_i g_i \right\|_\infty = \left\| (f_1 - f_0) - \sum_{i=1}^d (c_{1i} - c_{0i}) g_i \right\|_\infty \le \left\| f_1 - \sum_{i=1}^d c_{1i} g_i \right\|_\infty + \left\| f_0 - \sum_{i=1}^d c_{0i} g_i \right\|_\infty \le \varepsilon$$

$$\sum_{i=1}^d |c_i| \le \sum_{i=1}^d |c_{0i}| + \sum_{i=1}^d |c_{1i}| \le 4,$$

which confirms that $\mathcal{G}$ is a finite ε-basis of ReLU functions for $\Delta\mathcal{L}_{\text{cvx}}$ of size $\tilde{O}(1/\varepsilon^{2/3})$ and coefficient norm 4. □

Lipschitz Loss Functions. Every 1-Lipschitz loss function can be ε-approximated by a finite weighted sum of ReLU functions, but the coefficient norm of this approximation is not bounded by a constant (independent of ε). To achieve approximation with bounded coefficient norm, it is necessary to use threshold functions.

Definition 5.1. For $\theta \in [0, 1]$ let Th_θ denote the $\{\pm 1\}$-valued threshold function $\text{Th}_\theta(v) = \text{sgn}(v - \theta)$, with the convention that $\text{Th}_\theta(\theta) = 1$. For a subset $\Gamma \subseteq [0, 1]$ let $\text{Th}^\Gamma = \{\text{Th}_\theta \mid \theta \in \Gamma\}$. When $\Gamma(\varepsilon)$ is the set of integer multiples of ε in $[0, 1]$, we will abbreviate $\text{Th}^{\Gamma(\varepsilon)}$ as Th^ε. We will also abbreviate $\text{Th}^{[0,1]}$ as Th.

Lemma 5.4 (Threshold Basis for Lipschitz Functions)**.** For the class $\mathcal{L}_{\text{lip}}$ of 1-Lipschitz loss functions, the class $\mathcal{G} = \text{Th}^{\varepsilon/2}$ is a finite ε-basis for $\Delta\mathcal{L}_{\text{lip}}$ of size $\lceil \frac{2}{\varepsilon} + 1 \rceil$ and coefficient norm 4.

Proof of Lemma 5.4. To simplify notation we will prove Th^ε is a 2ε-basis for $\Delta\mathcal{L}_{\text{lip}}$ of size $\lceil \frac{1}{\varepsilon} + 1 \rceil$ and coefficient norm 2. (The lemma follows by reinterpreting ε in this proof as $\varepsilon/2$ in the lemma statement.) For $\ell \in \mathcal{L}_{\text{lip}}$ the function $f(p) = \Delta\ell(p) = \ell(p, 1) - \ell(p, 0)$ is 2-Lipschitz and $[-1, 1]$-valued. We can construct a piecewise constant function $\hat{f}$ by setting $\hat{f}(x) = f\left(\lfloor \frac{x}{\varepsilon} \rfloor \cdot \varepsilon\right)$ so that $|f(x) - \hat{f}(x)| \le 2\varepsilon$. We'll now express $\hat{f}$ as a linear combination of functions in Th^ε. Let $g_i(v)$ denote the function $\text{Th}_{i\varepsilon} \in \text{Th}^\varepsilon$.

$$\hat{f}(x) = \hat{f}(0) g_0(x) + \sum_{i \in [\lceil 1/\varepsilon \rceil]} (\hat{f}(i\varepsilon) - \hat{f}((i-1)\varepsilon)) g_i(x).$$

To see this, observe that we can simplify the RHS to

$$\hat{f}(0)(g_0(x) - g_1(x)) + \sum_{i \in [\lceil 1/\varepsilon \rceil]} \hat{f}(i\varepsilon)(g_i(x) - g_{i+1}(x))$$

Since $(g_i(x) - g_{i+1}(x))$ is 1 only when $i\varepsilon \le x < (i+1)\varepsilon$ and $\hat{f}(x)$ is constant in this interval, we get the desired result. Now we need to bound the coefficient norm. We know that $|\hat{f}(0)| \le 1$ and since f is 1-lipschitz, $|(\hat{f}(i\varepsilon) - \hat{f}((i-1)\varepsilon))| \le \varepsilon$ for all i. Thus, the coefficient norm is bounded by 2. □

Bounded Variation Losses and Proper Losses. Let $\mathcal{L}_{\text{BV}}$ denote the class of bounded variation losses, defined as follows.

Definition 5.2 (Bounded Variation)**.** A function $f : [0,1] \to \mathbb{R}$ has *bounded variation* if the quantity

$$V(f) = \sup\left\{ \sum_{i=1}^{n} |f(x_i) - f(x_{i-1})| \;\middle|\; 0 = x_0 < x_1 < \cdots < x_n = 1 \right\}$$

is finite. The class $\mathcal{L}_{\mathrm{BV}}$ of bounded variation losses consists of all loss functions $\ell(p, y)$ taking values in $[-1, 1]$ that satisfy $V(\Delta\ell) \le 2$.

Bounded variation losses are an extremely general family that includes all other loss classes considered in this paper.

Lemma 5.5. The class $\mathcal{L}_{\mathrm{BV}}$ includes all 1-Lipschitz losses, convex losses (with values in $[-1/4,\ 1/4]$, regardless of whether or not they are Lipschitz continuous), and proper losses taking values in $[-1, 1]$.

Proof. It is clear from the definition that $V(f) \le 2$ when f is monotone and $[-1, 1]$-valued, and also when f is 2-Lipschitz. Hence, the class $\mathcal{L}_{\mathrm{lip}}$ of 1-Lipschitz losses is in $\mathcal{L}_{\mathrm{BV}}$ (since $\Delta\ell$ is 2-Lipschitz whenever ℓ is 1-Lipschitz) and the class $\mathcal{L}_{\mathrm{proper}}$ of proper losses is in $\mathcal{L}_{\mathrm{BV}}$ (since $\Delta\ell$ is monotone whenever ℓ is proper). For a convex function f taking values in $[-1/4,\ 1/4]$, if x^* denotes a point in $[0, 1]$ where f attains its minimum, then f admits a representation of the form

$$f(x) = f(x^*) + f_0(x) + f_1(x)$$

where f_0, f_1 are monotone non-increasing and non-decreasing functions (respectively) from $[0, 1]$ to $[0, 1/2]$ satisfying $f_0(x) = 0$ for $x \ge x^*$ and $f_1(x) = 0$ for $x \le x^*$. From this representation it is clear that $V(f) \le V(f_0) + V(f_1) \le 1$. If $\ell(p, y)$ is a convex loss taking values in $[-1/4,\ 1/4]$ then $\Delta\ell$ is a difference of two convex $[-1/4,\ 1/4]$-valued functions, so $V(\Delta\ell) \le 2$. □

For bounded variation losses we have the following approximate basis.

Lemma 5.6. For any $\varepsilon > 0$, the class Th of all threshold functions on $[0, 1]$ is a ε-basis for $\Delta\mathcal{L}_{\mathrm{BV}}$ with sparsity $\lceil 2/\varepsilon + 1 \rceil$ and coefficient norm 3.

Proof. For $\ell \in \mathcal{L}_{\mathrm{BV}}$ let $f = \Delta\ell$, and consider the sequence $x_0 < x_1 < x_2 < \cdots < x_n$ defined inductively by setting $x_0 = 0$ and $x_{m+1} = \inf\{x > x_m \,:\, |f(x) - f(x_m)| > \varepsilon\}$ for all $m \ge 0$ such that the set in question is non-empty. The sequence ends with the first element x_n such that $\{x > x_n \,:\, |f(x) - f(x_n)| > \varepsilon\}$ is empty. For notational convenience we define $x_{n+1} = 1$. From the definition of $V(f)$ we see that $V(f) \ge n\varepsilon$, from which we deduce $n \le 2/\varepsilon$.

Define a piecewise-constant function $\hat{f} : [0, 1] \to [-1, 1]$ by setting $\hat{f}(x) = f(x_m)$ where x_m is the maximum element of $\{x_0, \ldots, x_{n+1}\} \cap [0, x]$. By construction, the inequality $|f(x) - \hat{f}(x)| \le \varepsilon$ holds for all $x \in [x_m, x_{m+1}]$, for all $m \in \{0, \ldots, n\}$. The union of the intervals $[x_m, x_{m+1}]$ equals $[0, 1]$, so $\|f - \hat{f}\|_\infty \le \varepsilon$.

For $i = 0, 1, \ldots, n$, let g_i denote the threshold function Th_{x_i}. As in the proof of Lemma 5.4 we have

$$\hat{f}(x) = \hat{f}(0)g_0(x) + \sum_{i=1}^{n+1} \left(\hat{f}(x_i) - \hat{f}(x_{i-1})\right) g_i(x).$$

The coefficient norm is bounded by $|\hat{f}(0)| + \sum_{i=1}^{n+1} |\hat{f}(x_i) - \hat{f}(x_{i-1}| \le 1 + V(f) \le 3$. □

The approximate basis Th is unfortunately not finite. This difficulty is inherent: the class $\Delta\mathcal{L}_{\text{proper}}$ includes, for each $\theta \in (0,1)$, a step function with a step of height 1 at θ. To approximate such a step function within ε in the ∞-norm, one must use a function having a jump discontinuity at θ, provided $\varepsilon < 1/2$. Hence, any ε-approximate basis for $\Delta\mathcal{L}_{\text{proper}}$ must include functions with jump discontinuities at every $\theta \in (0,1)$. However, this difficulty can be overcome for the class $\Delta\mathcal{L}_{\text{BV}} \circ \mathcal{H}$ whenever $\mathcal{H}$ is a class of hypothesis functions taking values in a finite subset of $[0,1]$.

Lemma 5.7. If $\mathcal{H}$ is a class of hypothesis functions taking values in a finite set Γ with $\{0,1\} \subseteq \Gamma \subset [0,1]$, then for all $\varepsilon > 0$ the class $\text{Th}^{\Gamma} \circ \mathcal{H}$ is a ε-basis for $\Delta\mathcal{L}_{\text{BV}} \circ \mathcal{H}$ with sparsity $\lceil 2/\varepsilon + 1 \rceil$ and coefficient norm 3.

Proof. From Lemma 4.8 and Lemma 5.6 we know that $\text{Th} \circ \mathcal{H}$ is a ε-basis for $\Delta\mathcal{L}_{\text{BV}} \circ \mathcal{H}$ with sparsity $\lceil 2/\varepsilon + 1 \rceil$ and coefficient norm 3. However, for every $\theta \in [0,1]$, if γ is the minimum element of $\Gamma \cap [\theta, 1]$ then $\text{Th}_\theta \circ h = \text{Th}_\gamma \circ h$ for every $h \in \mathcal{H}$. Hence, $\text{Th}^{\Gamma} \circ \mathcal{H} = \text{Th} \circ \mathcal{H}$ and the lemma follows. □

Here show a closure property for the class of all thresholds that will be useful for our omniprediction results.

Lemma 5.8 (Closure Property)**.** If $f, g : [-1,1] \to [-1,1]$ are monotone non-decreasing functions, then $f \circ g$ is a monotone non-decreasing function.

Proof of Lemma 5.8. Take any $x_1, x_2 \in [-1,1]$ such that $x_1 \le x_2$. Since g is monotone non-decreasing, it follows that

$$g(x_1) \le g(x_2).$$

Next, since f is monotone non-decreasing and $g(x_1) \le g(x_2)$, applying f to both sides preserves the order:

$$f(g(x_1)) \le f(g(x_2)).$$

Therefore, $f \circ g$ is monotone non-decreasing, as required. □

6 Offline Omniprediction

In this section, we show how the observation that proper calibration and multiaccuracy are sufficient for omniprediction leads to optimal sample complexity bounds for omniprediction in the offline setting. In particular, the algorithm we describe here will use a sample complexity that, for a given loss class $\mathcal{L}$, depends near-optimally on the sample complexity to cover $\Delta\mathcal{L} \circ \mathcal{H}$. In Section 6.1, we describe an online-to-batch procedure that outputs a *randomized* omnipredictor by running Algorithm 3 on i.i.d. samples from the distribution $\mathcal{D}$. In all, we establish the following result.

Theorem 6.1. There exists an sample-efficient algorithm $\mathcal{A}$ that for any distribution $\mathcal{D}$ supported on $\mathcal{X} \times \{0,1\}$, for any class of loss functions $\mathcal{L} \subseteq \mathcal{L}_{\text{BV}}$ that is $(\mathcal{G}, T^{-1/2})$-spanned with coefficient norm λ, any hypothesis class $\mathcal{H}$, and $\varepsilon > 0$, learns an $(\mathcal{L}, \mathcal{H}, \varepsilon)$-omnipredictor with the following properties:

- $\mathcal{A}$ returns a randomized omnipredictor that mixes over $\text{poly}(1/\varepsilon)$ postprocessed hypotheses from $\mathcal{H}$.

- $\mathcal{A}$ uses $m \le \tilde{O}(\lambda \cdot d_{\Delta\mathcal{G}\circ\mathcal{H}}/\varepsilon^2)$ samples drawn i.i.d. from $\mathcal{D}$, where $d_{\Delta\mathcal{G}\circ\mathcal{H}}$ denotes the VC dimension of $\Delta\mathcal{G}\circ\mathcal{H}$ or the fat-shattering dimension at scale ε in the case of real-valued class.

In particular, for any class $\mathcal{L} \subseteq \mathcal{L}_{\mathrm{BV}}$ of bounded-variation losses — including proper losses, convex losses, and 1-Lipschitz losses — the sample complexity of $(\mathcal{L}_{\mathrm{BV}}, \mathcal{H})$-omniprediction scales with the statistical complexity of $\mathrm{Th}\circ\mathcal{H}$, and the sample complexity of $(\mathcal{L}_{\mathrm{cvx}}, \mathcal{H})$-omniprediction scales with the statistical complexity of $\mathrm{ReLU}^{1/T}\circ\mathcal{H}$.

This theorem follows from Theorem 6.2, using Corollary 6.3 to instantiate the online weak agnostic learner.

6.1 Learning Randomized Omnipredictors via Online-to-Batch Conversion

To establish offline omnipredictors, we prove the following technical result.

Theorem 6.2. Let $\mathcal{H}$ be a class of hypothesis functions. Let $\mathcal{L}$ and $\mathcal{G}$ be classes of loss functions such that $\mathcal{L}$ is $(\mathcal{G}, \varepsilon)$-spanned with coefficient norm λ for some $\varepsilon \le \frac{1}{\sqrt{T}}$. Given a *online* weak agnostic learner for $\Delta\mathcal{G}\circ\mathcal{H}$, failure probability δ, and a sequence of T i.i.d samples $(x_1, y_1), \dots (x_T, y_T) \sim \mathcal{D}$, Algorithm 3 outputs a sequence of predictors $p_1, \dots, p_T$ such that the randomized predictor $\hat{\mathbf{p}} = \mathrm{unif}\{p_1, \dots, p_T\}$ satisfies

$$\mathop{\mathbf{E}}_{p\sim\hat{\mathbf{p}}}\mathop{\mathbf{E}}_{(x,y)\sim\mathcal{D}}[\ell(k_\ell(p(x)), y)] \le \min_{h\in\mathcal{H}}\mathop{\mathbf{E}}_{(x,y)\sim\mathcal{D}}[\ell(h(x), y)] + O\left(\lambda \cdot \mathsf{rad}_T(\Delta\mathcal{G}\circ\mathcal{H}) + \lambda\sqrt{\frac{\ln T/\delta}{T}}\right) + \frac{\lambda}{T}\cdot \mathrm{OracleReg}_{\Delta\mathcal{G}\circ\mathcal{H}}(T)$$

with probability $1-\delta$ over the randomness of the algorithm and the sampling from $\mathcal{D}$. Moreover, each returned predictor can be represented as $p_t = v_t \circ q_t$, where $q_1, \dots, q_T$ are the predictors returned by the online weak agnostic learner and $v_t : [-1, 1] \to [0, 1]$ is a post-processing function.

This theorem is the main result that we prove across this section. We break the proof into a series of technical lemmas. The lemmas serve to bound the online regret, which quantifies the error of $\hat{\mathbf{p}}$ at approximating the empirical statistics, and separately bound the generalization of the empirical statistics to their distributional quantities. First, we give the proof of the theorem assuming our subsequent lemmas, followed by technical sections to establish the lemmas.

Proof of Theorem 6.2. Since $\mathcal{L}$ is $(\mathcal{G}, \varepsilon)$-spanned with coefficient norm λ, then $\Delta\mathcal{L}$ is $(\Delta\mathcal{G}, \varepsilon)$-spanned with coefficient norm 2λ. In Theorem 4.1, we show that, given an online weak agnostic learner for $\Delta\mathcal{G}\circ\mathcal{H}$, Algorithm 3 outputs a sequence of predictions $p_1, \dots, p_T$ such that

$$\mathbf{E}[\mathrm{PCalErr}(\mathbf{p}, \mathbf{x}, \mathbf{y})] + \mathbf{E}[(\Delta\mathcal{G}\circ\mathcal{H})\text{-MAErr}(\mathbf{p}, \mathbf{x}, \mathbf{y})] \le O\left(\sqrt{T\ln T/\delta}\right) + \mathrm{OracleReg}_{\Delta\mathcal{G}\circ\mathcal{H}}(T)$$

and together with Lemma 4.9, this implies

$$\mathbf{E}[(\mathcal{L}, \mathcal{H})\text{-OmniRegret}(\mathbf{p}, \mathbf{x}, \mathbf{y})] \le O\left(\lambda\sqrt{T\ln T/\delta}\right) + \lambda\cdot\mathrm{OracleReg}_{\Delta\mathcal{G}\circ\mathcal{H}}(T)$$

That is, for all $\ell \in \mathcal{L}, h \in \mathcal{H}$,

$$\left[\frac{1}{T}\sum_{t=1}^{T}\ell(k_\ell(p_t(x_t)), y_t) - \frac{1}{T}\sum_{t=1}^{T}\ell(h(x_t), y_t)\right] \le O\left(\lambda\sqrt{\frac{\ln T/\delta}{T}}\right) + \frac{\lambda}{T}\cdot\mathrm{OracleReg}^{\delta}_{\Delta\mathcal{G}\circ\mathcal{H}}(T) \tag{47}$$

Now we would like to show that this also implies a bound on the expected omniprediction error under the true distribution. By Lemma 6.4 and Lemma 6.5, we know that with probability at least $1-\delta$,

$$\sup_{\ell\in\mathcal{L}} \left| \frac{1}{T}\sum_{t=1}^{T} \ell(k_\ell(p_t(x_t)), y_t) - \frac{1}{T}\sum_{t=1}^{T} \mathop{\mathbf{E}}_{(x,y)\sim\mathcal{D}}[\ell(k_\ell(p_t(x)), y)] \right| \le \frac{1}{T} + O\left(\sqrt{\frac{\ln\frac{T}{\delta}}{T}}\right)$$

since our algorithms output predictions that are multiples of $1/T$ and

$$\sup_{\ell\in\mathcal{L}, h\in\mathcal{H}} \left| \frac{1}{T}\sum_{t=1}^{T} \ell(h(x_t), y_t) - \mathop{\mathbf{E}}_{(x,y)\sim\mathcal{D}}[\ell(h(x), y)] \right| \le O\left(\lambda\cdot \mathsf{rad}_T(\Delta\mathcal{G}\circ\mathcal{H}) + \lambda\sqrt{\frac{\ln\frac{1}{\delta}}{T}}\right).$$

Plugging into the Equation (47), we obtain that with probability at least $1-\delta$,

$$\sup_{\ell\in\mathcal{L},\, h\in\mathcal{H}} \left[\frac{1}{T}\sum_{t=1}^{T} \mathop{\mathbf{E}}_{(x,y)\sim\mathcal{D}}[\ell(k_\ell(p_t(x), y)] - \mathop{\mathbf{E}}_{(x,y)\sim\mathcal{D}} \ell(h(x), y) \right] \tag{48}$$

$$\le O(\lambda\cdot \mathsf{rad}_T(\Delta\mathcal{G}\circ\mathcal{H})) + O\left(\lambda\sqrt{\frac{\ln T/\delta}{T}}\right) + \frac{\lambda}{T}\cdot \mathrm{OracleReg}^{\delta}_{\Delta\mathcal{G}\circ\mathcal{H}}(T) \tag{49}$$

The claim that each predictor p_t can be expressed as $v_t \circ q_t$ follows the guarantees from the Augmented Proper Calibration algorithm in Proposition 3.10. □

6.2 Implementing an Online Weak Agnostic Learner with Offline Sample Complexity

Online weak agnostic learning with i.i.d. features is a special case of the hybrid online learning problem (Lazaric and Munos, 2009; Wu et al., 2022). These prior results establish near-optimal dependence on the offline sample complexity of the class. In Corollary 6.3, we present this online weak agnostic learner, which is based on a multiplicative weights algorithm over a cover of size exponential in the VC (or fat-shattering) dimension of the class.

Corollary 6.3 (of Theorem 3 of Wu et al. (2022))**.** Let C be a class of hypothesis functions. Consider the setting where $x_1, \ldots, x_T$ are generated i.i.d. but revealed sequentially from a fixed distribution $\mathcal{D}$. There exists a multiplicative weights algorithm that returns a sequence of hypothesis $c_1, \ldots, c_T \in C$ such that for any sequence of $y_1, \cdots, y_T$ with probability at least $1-\delta$,

$$\sum_{t=1}^{T} c_t(x_t)\cdot y_t - \inf_{c\in C}\sum_{t=1}^{T} c(x_t)\cdot y_t \le \tilde{O}\left(\sqrt{T d_C}\right) + \tilde{O}\left(\sqrt{T\ln 1/\delta}\right).$$

where d_C represents the VC dimension of C in the case of a binary hypothesis class or the fat-shattering dimension at scale $1/\sqrt{T}$ in the case of a real-valued hypothesis class.

The proof of Corollary 6.3 is deferred to Appendix D.

6.3 Uniform Convergence Results for Online-to-Batch Conversion

Next, we establish the generalization bounds necessary for the empirical omniprediction statistics to converge to their distributional quantities.

Lemma 6.4 (Uniform Convergence for discretized predictors)**.** Let $p_1, \dots, p_T : \mathcal{X} \to \{0, \varepsilon, 2\varepsilon \dots, 1\}$ be a sequence of T predictors and let $\{(x_1, y_1), \dots, (x_T, y_T)\}$ be a sequence of i.i.d samples drawn from a distribution $\mathcal{D}$. Then for any class of loss functions $\mathcal{L}$, the following holds with probability at least $1 - \delta$,

$$\left| \frac{1}{T} \sum_{t=1}^{T} \ell(k_\ell(p_t(x_t)), y_t) - \frac{1}{T} \sum_{t=1}^{T} \mathop{\mathbf{E}}_{(x,y)\sim\mathcal{D}} [\ell(k_\ell(p_t(x)), y)] \right| \leq \varepsilon + O\left(\sqrt{\frac{\ln \frac{1}{\varepsilon\delta}}{T}} \right)$$

for all $\ell \in \mathcal{L}$

Proof of Lemma 6.4. Observe that it suffices to show that this holds for $\mathcal{L}_{\text{proper}}$. This is because, for any class of loss functions $\mathcal{L}$, the derived class $\{\ell \circ k_\ell : \ell \in \mathcal{L}\}$ is a subset of $\mathcal{L}_{\text{proper}}$. Recall that for every $\ell \in \mathcal{L}_{\text{proper}}$, there exists nonnegative coefficients $c_v(\ell)$ such that $\int_0^1 c_v(\ell) dv \leq 2$ and

$$\ell(p, y) = \int_0^1 c_v(\ell) \ell_v(p, y) dv$$

Consequently, it suffices to show that this holds for all ℓ_v for $v \in [0, 1]$. To see this observe that for any $\ell \in \mathcal{L}_{\text{proper}}$

$$\left| \frac{1}{T} \sum_{t=1}^{T} \ell(p_t(x_t), y_t) - \frac{1}{T} \sum_{t=1}^{T} \mathop{\mathbf{E}}_{(x,y)\sim\mathcal{D}} [\ell(p_t(x), y)] \right| \tag{50}$$

$$= \left| \frac{1}{T} \sum_{t=1}^{T} \int_0^1 c_v(\ell) \ell_v(p_t(x_t), y_t) dv - \frac{1}{T} \sum_{t=1}^{T} \mathop{\mathbf{E}}_{(x,y)\sim\mathcal{D}} \left[\int_0^1 c_v(\ell) \ell_v(p_t(x), y) dv \right] \right| \tag{51}$$

$$\leq \int_0^1 c_v(\ell) \left| \frac{1}{T} \sum_{t=1}^{T} \ell_v(p_t(x_t), y_t) - \frac{1}{T} \sum_{t=1}^{T} \mathop{\mathbf{E}}_{(x,y)\sim\mathcal{D}} [\ell_v(p_t(x), y)] \right| dv \tag{52}$$

Now we'll take advantage of the fact that $p_t(x)$ is in $\{0, \varepsilon, 2\varepsilon \dots, 1\}$ for all $x \in \mathcal{X}$. We'll show that for all $v \in [0, 1]$, there exists $v' \in \{0, \varepsilon, 2\varepsilon \dots, 1\}$ such that for all $x, y \in \mathcal{X} \times \mathcal{Y}$, $|\ell_v(p(x), y) - \ell_{v'}(p(x), y)| \leq \varepsilon$ for all v. In particular, $v' = \varepsilon \lceil \frac{v}{\varepsilon} \rceil$. We'll also adopt the convention that $\text{sgn}(0) = 1$. We'll first make the observation that because $p(x) \in \{0, \varepsilon, \dots, 1\}$, $\text{sgn}(v - p(x)) = \text{sgn}(v' - p(x))$ by our choice of v'. Completing the argument, we have that

$$\begin{aligned}
\ell_v(p(x), y) - \ell_{v'}(p(x), y) &= (y - v)\text{sgn}(v - p(x)) - (y - v')\text{sgn}(v' - p(x)) && (53)\\
&= (y - v)\text{sgn}(v - p(x)) - (y - v')\text{sgn}(v - p(x)) \\
&\qquad\qquad (\text{since } \text{sgn}(v - p(x)) = \text{sgn}(v' - p(x))) \\
&\leq (v - v')\text{sgn}(v - p(x)) && (54)\\
&\leq \varepsilon && (55)
\end{aligned}$$

Now we'll show that for all $v' \in \{0, \varepsilon, 2\varepsilon \dots, 1\}$,

$$\left| \frac{1}{T} \sum_{t=1}^{T} \ell_v(p_t(x_t), y_t) - \frac{1}{T} \sum_{t=1}^{T} \mathop{\mathbf{E}}_{(x,y)\sim\mathcal{D}} [\ell_v(p_t(x), y)] \right| \leq O\left(\sqrt{\frac{\ln \frac{1}{\varepsilon\delta}}{T}} \right)$$

Define $Z_t(v') = \ell_{v'}(p_t(x_t), y_t) - \mathbf{E}_{(x,y)\sim\mathcal{D}}[\ell_{v'}(p_t(x), y)]$. Observe that $Z_t(v')$ for $t = 1, \ldots, T$ forms a martingale difference sequence with bounded variance of 2. Thus, applying Azuma-Hoeffding's inequality, together with the union bound over all $v' \in \{0, \varepsilon, 2\varepsilon \ldots, 1\}$, we get that with probability at least $1 - \delta$,

$$\forall v' \in \{0, \varepsilon, 2\varepsilon \ldots, 1\} \quad \left| \frac{1}{T} \sum_{t=1}^{T} Z_t(v') \right| \leq O\left(\sqrt{\frac{\ln \frac{1}{\varepsilon\delta}}{T}} \right)$$

Concluding the proof, we have that for any $v \in [0, 1]$, with probability at least $1 - \delta$,

$$\left| \frac{1}{T} \sum_{t=1}^{T} \ell_v(p_t(x_t), y_t) - \frac{1}{T} \sum_{t=1}^{T} \mathop{\mathbf{E}}_{(x,y)\sim\mathcal{D}} [\ell_v(p_t(x), y)] \right| \leq \varepsilon + O\left(\sqrt{\frac{\ln \frac{1}{\varepsilon\delta}}{T}} \right)$$

Consequently, for any $\ell \in \mathcal{L}_{\text{proper}}$,

$$\left| \frac{1}{T} \sum_{t=1}^{T} \ell(p_t(x_t), y_t) - \frac{1}{T} \sum_{t=1}^{T} \mathop{\mathbf{E}}_{(x,y)\sim\mathcal{D}} [\ell(p_t(x), y)] \right| \tag{56}$$

$$\leq \int_0^1 c_v(\ell) \left| \frac{1}{T} \sum_{t=1}^{T} \ell_v(p_t(x_t), y_t) - \frac{1}{T} \sum_{t=1}^{T} \mathop{\mathbf{E}}_{(x,y)\sim\mathcal{D}} [\ell_v(p_t(x), y)] \right| dv \tag{57}$$

$$\leq 2\varepsilon + O\left(\sqrt{\frac{\ln \frac{1}{\varepsilon\delta}}{T}} \right) \tag{58}$$

Since $\ell \circ k_\ell$ is in $\mathcal{L}_{\text{proper}}$, this completes the result. □

Lemma 6.5 (Uniform Convergence for $\mathcal{G}$-spanning loss classes)**.** Let $\mathcal{L}$ be a $(\mathcal{G}, \varepsilon)$-spanned class of loss functions with coefficient norm λ for some $\varepsilon \leq T^{-1/2}$. Let $\{(x_1, y_1), \ldots, (x_T, y_T)\}$ be a sequence of i.i.d. samples drawn from a distribution $\mathcal{D}$. Then for a hypothesis class $\mathcal{H}$, the following holds with probability at least $1 - \delta$,

$$\left| \frac{1}{T} \sum_{t=1}^{T} \ell(h(x_t), y_t) - \mathop{\mathbf{E}}_{(x,y)\sim\mathcal{D}} [\ell(h(x), y)] \right| \leq O\left(\lambda \cdot \mathsf{rad}_T(\mathcal{G} \circ \mathcal{H})\right) + O\left(\lambda \sqrt{\frac{\ln \frac{1}{\delta}}{T}} \right)$$

uniformly for every $\ell \in \mathcal{L}$ and $h \in \mathcal{H}$.

Proof of Lemma 6.5. Since every $\ell \in \mathcal{L}$ is ε-close to the span of $\mathcal{G} \circ \mathcal{H}$, the proof will show that uniform convergence extends from $\mathcal{G} \circ \mathcal{H}$ to all losses $\ell \in \mathcal{L}$ and hypotheses $h \in \mathcal{H}$. In particular, standard application of uniform convergence results (Chapter 26 of Shalev-Shwartz and Ben-David (2014)) show that the following holds with probability at least $1 - \delta$:

$$\forall g \circ h \in \mathcal{G} \circ \mathcal{H},\ z \in \{0, 1\} \quad \left| \frac{1}{T} \sum_{t=1}^{T} g(h(x_t), z) - \mathop{\mathbf{E}}_{(x,y)\sim\mathcal{D}} [g(h(x), z)] \right| \leq \mathsf{rad}_T(\mathcal{G} \circ \mathcal{H}) + O\left(\sqrt{\frac{\ln \delta^{-1}}{T}} \right) \tag{59}$$

and

$$\forall g \circ h \in \mathcal{G} \circ \mathcal{H}\, z \in \{0,1\} \quad \left| \frac{1}{T} \sum_{t=1}^{T} y_t \cdot g(h(x_t), z) - \mathop{\mathbf{E}}_{(x,y)\sim\mathcal{D}} [y \cdot g(h(x), z)] \right| \leq \mathsf{rad}_T(\mathcal{G} \circ \mathcal{H}) + O\left(\sqrt{\frac{\ln \delta^{-1}}{T}} \right) \tag{60}$$

By definition of $(\mathcal{G}, \varepsilon)$-spanning, we know that for every $\ell \in \mathcal{L}$, there exists a signed measure $c_\ell(\omega)$ such that $\|c_\ell\| \leq \lambda$ and such that the function $\hat{\ell} : [0,1] \times \{0,1\} \to \mathbb{R}$ defined by

$$\hat{\ell}(p, y) = \int_\Omega g_\omega(p, y)\, \mathrm{d}c_\ell(\omega)$$

satisfies $\|\ell - \hat{\ell}\|_\infty \leq \varepsilon$. Then it follows that for every $\ell \in \mathcal{L}$ and $h \in \mathcal{H}$

$$\begin{aligned}
\left| \frac{1}{T} \sum_{t=1}^{T} \ell(h(x_t), y_t) - \mathop{\mathbf{E}}_{(x,y)\sim\mathcal{D}} [\ell(h(x), y)] \right| &\leq 2\varepsilon + \left| \frac{1}{T} \sum_{t=1}^{T} \hat{\ell}(h(x_t), y_t) - \mathop{\mathbf{E}}_{(x,y)\sim\mathcal{D}} [\hat{\ell}(h(x), y)] \right| \\
&= 2\varepsilon + \left| \int_\Omega \left(\frac{1}{T} \sum_{t=1}^{T} g_\omega(h(x_t), y_t) - \mathop{\mathbf{E}}_{(x,y)\sim\mathcal{D}} [g_\omega(h(x), y)] \right) \mathrm{d}c_\ell(\omega) \right| \\
&\leq 2\varepsilon + \int_\Omega \left| \frac{1}{T} \sum_{t=1}^{T} g_\omega(h(x_t), y_t) - \mathop{\mathbf{E}}_{(x,y)\sim\mathcal{D}} [g_\omega(h(x), y)] \right| \mathrm{d}|c_\ell|(\omega). \\
&\leq 2\varepsilon + \sup_{g \in \mathcal{G}} \left| \frac{1}{T} \sum_{t=1}^{T} g(h(x_t), y_t) - \mathop{\mathbf{E}}_{(x,y)\sim\mathcal{D}} [g(h(x), y)] \right| \cdot \|c_\ell\|. \qquad (61)
\end{aligned}$$

Observe that for any $g : [0,1] \times \{0,1\} \to \mathbb{R}$ and all $(p, y) \in [0,1] \times \{0,1\}$ we have

$$g(p, y) = g(p, 0) + yg(p, 1) - yg(p, 0).$$

Hence, assuming (59) and (60) hold, we have that for all $g \in \mathcal{G}$,

$$\begin{aligned}
\left| \frac{1}{T} \sum_{t=1}^{T} g(h(x_t), y_t) - \mathop{\mathbf{E}}_{(x,y)\sim\mathcal{D}} [g(h(x), y)] \right| &\leq \left| \frac{1}{T} \sum_{t=1}^{T} g(h(x_t), 0) - \mathop{\mathbf{E}}_{(x,y)\sim\mathcal{D}} [g(h(x), 0)] \right| \\
&\quad + \left| \frac{1}{T} \sum_{t=1}^{T} y_t \cdot g(h(x_t), 1) - \mathop{\mathbf{E}}_{(x,y)\sim\mathcal{D}} [y \cdot g(h(x), 1)] \right| \\
&\quad + \left| \frac{1}{T} \sum_{t=1}^{T} y_t \cdot g(h(x_t), 0) - \mathop{\mathbf{E}}_{(x,y)\sim\mathcal{D}} [y \cdot g(h(x), 0)] \right| \\
&\leq 3\, \mathsf{rad}_T(\mathcal{G} \circ \mathcal{H}) + O\left(\sqrt{\frac{\ln \delta^{-1}}{T}} \right).
\end{aligned}$$

Substituting this bound back into inequality (61), we obtain

$$\begin{aligned}
\left| \frac{1}{T} \sum_{t=1}^{T} \ell(h(x_t), y_t) - \mathop{\mathbf{E}}_{(x,y)\sim\mathcal{D}} [\ell(h(x), y)] \right| &\leq 2\varepsilon + \left[3\mathsf{rad}_T(\mathcal{G} \circ \mathcal{H}) + O\left(\sqrt{\frac{\ln \delta^{-1}}{T}} \right) \right] \|c_\ell\| \qquad (62) \\
&\leq 2\varepsilon + 3\lambda \mathsf{rad}_T(\mathcal{G} \circ \mathcal{H}) + O\left(\lambda \sqrt{\frac{\ln \delta^{-1}}{T}} \right) \qquad (63)
\end{aligned}$$

and the result follows by our assumption that $\varepsilon \leq T^{-1/2}$. □

7 Oracle-Efficient Offline Omniprediction

In this final section, we describe an algorithm that achieves omniprediction in the offline setting, returning predictors with efficient representations, in near-optimal sample complexity, using only an offline ERM oracle. Our construction is modeled after previous algorithms, but critically, we replace the OWAL oracle with a new abstraction that we call a Distributional Online Weak Agnostic Learner. We give an efficient implementation of this oracle given access to an ERM oracle, via a Frank-Wolfe reduction to solve an entropy-regularized ERM problem. except that we replace the online weak agnostic learner with a distributional online weak agnostic learner, which we efficiently implement using the offline ERM oracle.

To begin, we introduce the new hybrid learning oracle.

Definition 7.1. A Distributional Online Weak Agnostic Learner (DOWAL) is an online algorithm initialized with a hypothesis class C, a labeling function class $\mathcal{R}$, and a failure probability δ. It is initialized with a dataset of samples $\{(x_t, y_t)\}_{t=1}^{T}$. Each timestep t:

1. The algorithm outputs a hypothesis function $q_t : \mathcal{X} \to [-1, 1]$
2. The algorithm receives an adversarial labeling function $r_t : \mathcal{X} \times \mathcal{Y} \to [-1, 1] \in \mathcal{R}$

We say an algorithm implements a DOWAL with regret bound $\text{DistOracleReg}_{C,\mathcal{R}}^{m,\delta}(T)$ if given an initial dataset of samples $(x_i, y_i)_{i=1}^{m}$ drawn i.i.d. from a fixed distribution $\mathcal{D}$, it guarantees with probability at least $1 - \delta$

$$\max_{c \in C} \sum_{t=1}^{T} \mathop{\mathbf{E}}_{\mathcal{D}}[c(x) r_t(x, y)] \;\le\; \sum_{t=1}^{T} \mathop{\mathbf{E}}_{\mathcal{D}}[q_t(x) r_t(x, y)] \;+\; \text{DistOracleReg}_{C,\mathcal{R}}^{m,\delta}(T).$$

For the DOWAL used in this paper, the relationship between the hypothesis class C and the labeling function class $\mathcal{R}$ is generally as follows.

$$\mathcal{R} = \{r : \mathcal{X} \times \mathcal{Y} \to [-1, 1] \;\mid\; r(x, y) = y - u(c(x)),\, c \in C, u : [-1, 1] \to [0, 1]\}. \tag{64}$$

We will use the term "DOWAL for C" to refer to a DOWAL for a pair $(C, \mathcal{R})$ satisfying the relationship in Equation (64).

Finally, we say a DOWAL is *proper* if its outputs $q_1, \ldots, q_T$ belong to the class C. We may denote the output sequence of a proper DOWAL by $c_1, \ldots, c_T$ rather than $q_1, \ldots, q_T$ to emphasize that the outputs belong to the function class C.

Using this oracle, we can describe our novel approach to distributional omniprediction. The resulting algorithm, Algorithm 5, is an oracle-efficient implementation of our algorithmic framework, tailored to the distributional setting. As before, we invoke the distributional OWAL with the multiaccuracy class $C = \Delta\mathcal{L} \circ \mathcal{H}$, but here, we use a label class $\mathcal{R}$ to encompass the residual functions $y - p_t(x)$. Critically, to prove the DOWAL guarantees, we need to ensure that the prediction functions $\{p_t\}$ live in some class of functions about which we can reason.

Theorem 7.1. Algorithm 5 is an oracle-efficient algorithm that for any distribution $\mathcal{D}$ supported on $\mathcal{X} \times \{0, 1\}$, any class of loss functions $\mathcal{L} \subseteq \mathcal{L}_{\text{BV}}$, and any hypothesis class $\mathcal{H}$, learns an $(\mathcal{L}, \mathcal{H})$-omnipredictor given $2T$ i.i.d. samples from $\mathcal{D}$. The algorithm has the following properties:

Algorithm 5 Oracle-Efficient Offline Omniprediction Using DOWAL

procedure OFFLINEOMNIPREDICTION(T)
  Input: A dataset of $2T$ samples $\{(x_t, y_t)\}_{t=1}^{2T}$, drawn i.i.d. from $\mathcal{D}$.
  Input: A failure probability parameter, $\delta > 0$.
  Split the samples into two equal-sized sets, $D_{\text{apcal}} = \{(x_t, y_t) \mid t \le T\}$ and $D_{\text{dowal}} = \{(x_t, y_t) \mid t > T\}$.
  Input: Initialize Algorithm 6 instance, $\mathcal{A}_{\text{dowal}}$, using function class $C = \{+1, -1\} \cdot \text{Th} \circ \mathcal{H}$ and samples D_{dowal}.
  Input: Initialize Algorithm 2 instance, $\mathcal{A}_{\text{apcal}}$, with failure probability δ
  Receive c_1 from $\mathcal{A}_{\text{dowal}}$
  for $t = 1$ to T **do**
    Send c_t to $\mathcal{A}_{\text{apcal}}$
    Receive p_t from $\mathcal{A}_{\text{apcal}}$
    Send $(x_t, y_t) \in D_{\text{apcal}}$ to $\mathcal{A}_{\text{apcal}}$
    Send labeling function $r_t(x, y) = y - p_t(x)$ to $\mathcal{A}_{\text{dowal}}$
    Receive c_{t+1} from $\mathcal{A}_{\text{dowal}}$
  Output randomized predictor $\hat{\mathbf{p}} = \text{unif}(p_1, \ldots, p_T)$.

- It returns a randomized omnipredictor that mixes over T predictors, each represented as $p_t = v_t \circ h_t$, where $h_t \in \mathcal{H}$ and $v_t : [-1, 1] \to [0, 1]$ is a postprocessing function.
- It is oracle-efficient, making $O(T^{5/2})$ calls to an *offline* ERM oracle for $\text{Th} \circ \mathcal{H}$.
- The error of the omnipredictor scales near-optimally with the *offline* Rademacher complexity: $\varepsilon(T) = O\left(\log^{3/2}(T) \cdot \mathsf{rad}_T(\text{Th} \circ \mathcal{H})\right) + O\left(\sqrt{\ln(T/\delta)/T}\right)$.

We begin with a high-level proof that invokes technical lemmas proved subsequently. We justify the claims on representation complexity, oracle-efficiency, and statistical complexity.

Proof of Theorem 7.1. The claim about the omnipredictor's representation complexity is justified by the following observations.

1. By Lemma 7.4, $\mathcal{A}_{\text{dowal}}$ is a proper DOWAL, returning functions c_t in $C = \{+1, -1\} \cdot \text{Th} \circ \mathcal{H}$.
2. By Proposition 3.10, $\mathcal{A}_{\text{apcal}}$ returns predictors p_t in $U \circ C$, where U is the class of all functions from $[-1, 1]$ to $[0, 1]$.
3. Hence, p_t belongs to the class $U \circ (\pm\text{Th}) \circ \mathcal{H}$. In other words, $p_t = v_t \circ h_t$ where $h_t \in \mathcal{H}$ and $v_t \in U \circ (\pm\text{Th})$.

The claim about oracle-efficiency is justified by the following observations.

1. Algorithm 6 is called T times. Each call involves solving an entropy-regularized ERM problem using Algorithm 7 with parameters $m = T$ and $\varepsilon = \eta = T^{-1/2}$.
2. By Lemma 7.6, each call to Algorithm 7 makes $O(m/\varepsilon) = O(T^{3/2})$ calls to the ERM oracle for C.

3. An ERM for $C = \{+1, -1\} \cdot \mathrm{Th} \circ \mathcal{H}$ can be implemented using two ERM calls for $\mathrm{Th} \circ \mathcal{H}$: one with the given coefficient vector and one with the negation of that vector.

The claim about omniprediction error is justified using Lemma 4.4 and Proposition 4.5 of Gopalan et al. (2023), which decomposes $(\mathcal{L}, \mathcal{H})$-omniprediction error as the sum of $\mathcal{L}$-Decision-OI error and $\Delta\mathcal{L} \circ \mathcal{H}$-multiaccuracy error. The former is bounded by $\mathrm{PCalErr}_{\mathcal{D}}(\hat{\mathbf{p}})$ according to Theorem 3.1. The latter is bounded by $C\text{-MAErr}_{\mathcal{D}}(\hat{\mathbf{p}})$ according to Lemmas 4.12 and 5.6. Thus, to justify the stated upper bound on omniprediction error, it suffices to bound the distributional proper calibration error and $\mathrm{Th} \circ \mathcal{H}$-multiaccuracy error of $\hat{\mathbf{p}}$.

The bound $\mathrm{PCalErr}_{\mathcal{D}}(\hat{\mathbf{p}}) = O(\sqrt{T \log T/\delta})$ is a direct consequence of Proposition 3.10. Note that the splitting of the sample set into D_{apcal} and D_{dowal} is vital here. From the standpoint of the notation in Section 3.4, the samples $\{(x_t, y_t)\}_{t=T+1}^{2T}$ that constitute D_{dowal} are part of the adversary's random seed r_{adv}. Thus, the fact that q_t may depend on all of the samples in D_{dowal} is consistent with the online restrictions placed on the APCAL and its adversary by Proposition 3.10.

The bound on C-multiaccuracy error of $\hat{\mathbf{p}}$ will follow by quantifying the oracle regret of the DOWAL implemented by Algorithm 6. Recall from Lemma 7.4 that the algorithm with m samples and failure probability δ satisfies the oracle regret bound

$$\mathrm{DistOracleReg}_{C,\mathcal{R}}^{m,\delta}(T) = O\left(\sqrt{T \log 1/\delta}\right) + O(T \cdot \mathsf{rad}_T(C \cdot \mathcal{R})).$$

Below, in Lemma 7.2 we prove the Rademacher complexity bound

$$\mathsf{rad}_T(C \cdot \mathcal{R}) \in O(\log^{3/2}(T) \cdot \mathsf{rad}_T(\mathrm{Th} \circ \mathcal{H})). \tag{65}$$

Hence,

$$\begin{aligned}
C\text{-MAErr}_{\mathcal{D}}(\hat{\mathbf{p}}) &= \max_{c \in C}\left\{\frac{1}{T}\sum_{t=1}^{T} \mathop{\mathbf{E}}_{\mathcal{D}}[c(x)(y - p_t(x))]\right\} \\
&= \max_{c \in C}\left\{\frac{1}{T}\sum_{t=1}^{T} \mathop{\mathbf{E}}_{\mathcal{D}}[c(x) r_t(x, y)]\right\} && \text{by our choice of } r_t \\
&\le \frac{1}{T}\sum_{t=1}^{T} \mathop{\mathbf{E}}_{\mathcal{D}}[c_t(x) r_t(x, y)] \;+\; O\left(\sqrt{\tfrac{\ln(1/\delta)}{T}}\right) + O(\mathsf{rad}_m(C \cdot \mathcal{R})) && \text{by Lemma 7.4} \\
&= \frac{1}{T}\sum_{t=1}^{T} \mathop{\mathbf{E}}_{\mathcal{D}}[c_t(x)(y - p_t(x))] \;+\; O\left(\sqrt{\tfrac{\ln(1/\delta)}{T}}\right) + O(\mathsf{rad}_m(C \cdot \mathcal{R})) \\
&\le O\left(\sqrt{\tfrac{\ln(T/\delta)}{T}}\right) + O(\log^{3/2}(T) \cdot \mathsf{rad}_T(\mathrm{Th} \circ \mathcal{H})) && \text{by Proposition 3.10 and Eq. (65)}
\end{aligned}$$

which justifies the stated omniprediction error bound $\varepsilon(T)$. □

Next, we prove Lemma 7.2, which establishes the generalization bounds invoked in the proof of Theorem 7.1. This generalization bound, while relatively short to prove, is a critical and delicate part of our argument. In particular, to establish this lemma, we need to leverage specific properties of the class $C = \{+1, -1\} \cdot \mathrm{Th} \circ \mathcal{H}$ and the resulting $\mathcal{R}$ implied by Algorithm 5. This dependence on $\mathrm{Th} \circ \mathcal{H}$ leads to the error scaling by the corresponding Rademacher complexity (even for simpler loss classes than $\mathcal{L}_{\mathrm{BV}}$).

Lemma 7.2. Let $\mathcal{H} \subseteq \{h : \mathcal{X} \to [-1,1]\}$ be a class of hypothesis functions. Let C denote the class of thresholds on $\mathcal{H}$ i.e $C = \{+1,-1\} \cdot \mathrm{Th} \circ \mathcal{H}$. Let U denote the class of all functions $[-1,1] \to [0,1]$. Let $\mathcal{R}$ be the class of bivariate functions $r(x,y) = y - u(c(x))$ for some $u \in U, c \in C$. Then

$$\mathsf{rad}_m(C \cdot \mathcal{R}) \in O(\log^{3/2}(m) \cdot \mathsf{rad}_m(\mathrm{Th} \circ \mathcal{H}))$$

Proof. Recall that the (empirical) Rademacher complexity of a class $\mathcal{F}$ of real-valued functions is defined as

$$\hat{\mathsf{rad}}_m(\mathcal{F}) \;=\; \mathbb{E}_{\boldsymbol{\sigma}}\Big[\sup_{f \in \mathcal{F}} \frac{1}{m} \sum_{i=1}^{m} \sigma_i f(x_i)\Big],$$

where $\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_m)$ is a vector of i.i.d. Rademacher random variables, each taking values in $\{-1,+1\}$ with probability $1/2$.

By definition, every element of $C \cdot \mathcal{R}$ can be written as $c'(x)(y - u(c(x)))$ for some $c', c \in C$ and $u \in U$. Since $c(x) \in \{-1,+1\}$, we only need to consider how u maps the two discrete inputs -1 and $+1$ into $[0,1]$. Notice that

$$u(c(x)) \;=\; \frac{u(-1)+u(+1)}{2} \;+\; \frac{u(+1)-u(-1)}{2}\, c(x).$$

Thus, $c'(x)(y - u(c(x)))$ can be expanded to $c'(x)y - \alpha c'(x) - \beta c(x)c'(x)$ for some constants $\alpha, \beta \in [-1,1]$. This means that

$$\mathsf{rad}_m(C \cdot \mathcal{R}) \le \mathsf{rad}_m\left(\{c(x) \cdot y : c \in C\}\right) + \mathsf{rad}_m\left(\{c(x) : c \in C\}\right) + \mathsf{rad}_m\left(\{c(x)c'(x) : c, c' \in C\}\right) \tag{66}$$

We know that $\mathsf{rad}_m\left(\{\mathrm{Th}_\theta(h(x)) \cdot y : \theta \in [-1,1], h \in \mathcal{H}\}\right) \le \mathsf{rad}_m(\mathrm{Th} \circ \mathcal{H})$. By Lemma 7.3, we have that $\mathsf{rad}_m\left((\mathrm{Th} \circ \mathcal{H}) \cdot (\mathrm{Th} \circ \mathcal{H})\right) \le \log^{3/2}(T) \cdot \mathsf{rad}_m(\mathrm{Th} \circ \mathcal{H})$. Thus, we obtain the desired result. □

Lemma 7.3 (Corollary 6 of Rakhlin et al. (2015))**.** For a fixed binary function $b : \{\pm 1\}^k \to \{\pm 1\}$ and classes $G_1, \ldots, G_k$ of $\{\pm 1\}$-valued functions,

$$\mathsf{rad}_T(b(G_1, \ldots, G_k)) \le O(\log^{3/2}(T)) \sum_{j=1}^{k} \mathsf{rad}_T(G_j)$$

A tighter version of this claim holds for Gaussian complexities without the additional log factors; see Theorem 16 of Bartlett and Mendelson (2003).

7.1 Implementing a DOWAL using FTRL

This subsection presents and analyzes a DOWAL that makes use of a subroutine called an *entropy-regularized ERM oracle over* C, defined as follows.

Definition 7.2. An entropy-regularized ERM oracle is initialized with a class of functions $C : \mathcal{X} \to [-1,1]$ and an approximation parameter ε. The oracle takes, as input, a set of pairs $(x_1, w_1), \ldots, (x_m, w_m) \in \mathcal{X} \times \mathbb{R}$. It outputs an explicit convex combination of elements of C, i.e. a sequence of hypotheses $c_1, \ldots, c_k \in C$ and coefficients $\alpha_1, \ldots, \alpha_k \in [0,1]$ summing up to 1, such that the weighted average $c = \sum_{i=1}^{k} \alpha_i c_i$ minimizes (within ε) the function $\sum_{i=1}^{m} c(x_i)w_i \;+\; \sum_{i=1}^{m} (c(x_i)+2)\log(c(x_i)+2)$.

We use $c(x_i) + 2$ rather than $c(x_i)$ in the regularizer because $c(x_i)$ takes values in $[-1, 1]$, whereas $\log(z)$ is only defined when z is strictly positive.

In Section 7.2 below, we show how to use the Frank-Wolfe method to implement an ε-approximate regularized ERM oracle using poly(m) calls to a standard ERM oracle.

Overview of Algorithm 6 The algorithm implements a DOWAL using the Follow The Regularized Leader (FTRL) approach. At each timestep t, we call the entropy-regularized ERM oracle using a dataset that labels the points $x_1, \ldots, x_m \in \mathcal{X}$ with labels $w_1, \ldots, w_m$ derived from the past adversarial functions $\{r_1, \ldots, r_{t-1}\}$. The oracle finds an approximate minimizer of the cumulative regularized empirical loss. It outputs this approximate minimizer represented as a convex combination of elements of C. To select the predictor c_t at each step, we sample a single function from C according to the convex combination's weights. This guarantees that $c_t \in C$, i.e. the algorithm is a *proper* DOWAL.

Algorithm 6 Distributional Online Weak Agnostic Learner using Exponentiated Gradient

Require: i.i.d. samples $S = \{(x_i, y_i)\}_{i=1}^m \sim \mathcal{D}^m$, time horizon T, failure probability δ
Ensure: Sequence of predictors $\{c_t\}_{t=1}^T$ where each $c_t \in C$

1: Set $\eta \leftarrow \sqrt{1/T}$
2: Initialize distribution over C by calling the entropy-regularized ERM oracle (in Algorithm 7):

$$\tilde{c}_1 \leftarrow \arg\min^{\eta}_{c \in C} \left\{ \sum_{i=1}^{m} (c(x_i) + 2) \log(c(x_i) + 2) \right\}.$$

Obtain from the oracle the convex combination $\{c_j, \alpha_j\}$ representing $\tilde{c}_1$.
3: **for** $t = 1$ to T **do**
4: Let $\{(c_j, \alpha_j)\}$ be the current convex combination representing the $\tilde{c}_t$.
5: Draw $c_t \in C$ by sampling according to weights $\{\alpha_j\}$.
6: Output c_t.
7: Receive adversary function $r_t \in \mathcal{R}$.
8: Update the convex combination by calling the entropy-regularized ERM oracle (in Algorithm 7):

$$\tilde{c}_{t+1} \in \arg\min^{\eta}_{c \in \mathrm{conv}(C)} \left\{ -\eta \sum_{s=1}^{t} \sum_{i=1}^{m} c(x_i) r_s(x_i, y_i) \; + \; \sum_{i=1}^{m} (c(x_i) + 2) \log(c(x_i) + 2) \right\}$$

9: Obtain from the oracle the updated convex combination $\{(c'_j, \alpha'_j)\}$ representing $\tilde{c}_{t+1}$.
10: **return:** Sequence of predictors $\{c_t\}_{t=1}^T$

Lines 2 and 8 of the algorithm use the notation $\arg\min^{\eta}$. This denotes the set of all points where the indicated function attains a value within η of its global minimum.

Lemma 7.4 (Exponentiated Gradient DOWAL)**.** Given m i.i.d samples from $\mathcal{D}$, an entropy-regularized ERM oracle over C (implemented in Algorithm 7) and failure probability $\delta > 0$, Algorithm 6 implements a high probability distributional DOWAL using a dataset of size m and guarantees the following with probability at

least $1-\delta$:

$$\max_{c\in C}\sum_{t=1}^{T}\mathop{\mathbf{E}}_{\mathcal{D}}[c(x)r_t(x,y)] \le \sum_{t=1}^{T}\mathop{\mathbf{E}}_{\mathcal{D}}[c_t(x)r_t(x,y)] + O\left(\sqrt{T\ln 1/\delta}\right) + O(T\cdot \mathsf{rad}_m(C\cdot\mathcal{R}))$$

where $C\cdot\mathcal{R}$ denotes the set of all functions on $\mathcal{X}\times\mathcal{Y}$ of the form $g(x,y)=c(x)r(x,y)$ for $c\in C$, $r\in\mathcal{R}$. The algorithm runs in time $O(m)$ per timestep, making $O(1)$ calls to the regularized ERM oracle. Moreover, it implements a *proper* DOWAL: each c_t chosen by the algorithm is in C.

Proof of Lemma 7.4. The proof follows the regret analysis of Follow-the-Regularized-Leader (FTRL) under entropy regularization combined with standard Rademacher generalization arguments, applied with high probability.

For each $c\in C$ define $u(c)=(c(x_1),c(x_2),\ldots,c(x_m))$ and let $\mathcal{K}$ denote the convex hull of the vectors $\{u(c)\mid c\in C\}$. Define the regularizer

$$\psi(u)=\sum_{i=1}^{m}(u_i+2)\log(u_i+2)$$

and the regularized objective function

$$g_t(u)=-\eta\sum_{s=1}^{t-1}\sum_{i=1}^{m}u_i r_s(x_i,y_i)+\psi(u)$$

and let

$$u_t\in\arg\min_{u\in\mathcal{K}}\{g_t(u)\}.$$

Let $\nabla_t\in\mathbb{R}^m$ denote the vector whose i^{th} coordinate is $-r_t(x_i,y_i)$. The standard analysis of FTRL, Lemma 7.5, shows that for all $u^*\in\mathcal{K}$:

$$\sum_{t=1}^{T}\langle\nabla_t,u_t-u^*\rangle\le 2\eta\sum_{t=1}^{T}\|\nabla_t\|_t^{*2}+\frac{\psi(u^*)-\psi(u_1)}{\eta}. \tag{67}$$

Here, the norm $\|\cdot\|_t^*$ denotes the dual norm defined by the regularizer ψ at u_t, i.e. $\|\nabla_t\|_t^*=\|\nabla_t\|_{\nabla^{-2}\psi(u_t)}$. For the entropy regularizer $\psi(u)$, the inverse Hessian matrix $\nabla^2\psi(u)$ is a diagonal matrix whose i^{th} diagonal entry is $\frac{1}{u_i+2}$. Since $\mathcal{K}\subseteq[-1,1]^m$, we have $1\le u_i+2\le 3$ for any $u\in\mathcal{K}$. This implies

$$\|\nabla_t\|^2_{\nabla^{-2}\psi(u_t)}\le 3\|\nabla_t\|_2^2\le 3m, \tag{68}$$

where the last inequality follows because each of the m coordinates of ∇_t belongs to the interval $[-1,1]$.

The regularizer ψ attains values between 0 and $m\log(3)$, so $\frac{\psi(u^*)-\psi(u_1)}{\eta}\le\frac{m\log(3)}{\eta}$. Substituting this bound and the bound (68) into (67), we obtain

$$\sum_{t=1}^{T}\langle\nabla_t,\,u_t\rangle\;-\;\sum_{t=1}^{T}\langle\nabla_t,\,u^*\rangle\le 6\eta mT+\frac{m\log(3)}{\eta}\le 8m\sqrt{T} \tag{69}$$

by our choice of $\eta=\sqrt{1/T}$.

Next we bound the regret of playing an η-approximate minimizer of $g_t(u)$ in each step, rather than the exact minimizer. Observe that $g_t(u)$ is a linear function of u plus $\psi(u)$, so $\nabla^2 g_t(u) = \nabla^2\psi(u)$. Earlier we calculated that for all $u \in \mathcal{K}$, the Hessian matrix $\nabla^2\psi(u)$ is a diagonal matrix with entries between $\frac{1}{3}$ and 1, hence g_t is $\left(\frac{1}{3}\right)$-strongly convex. Since u_t is the global minimizer of g_t, it follows from strong convexity that

$$\begin{aligned}
\tfrac{1}{6}\|u(\tilde{c}_t) - u_t\|_2^2 &\le g_t(u(\tilde{c}_t) - g_t(u_t) \le \eta \\
\|u(\tilde{c}_t) - u_t\|_2^2 &\le 6\eta \\
\langle \nabla_t,\, u(\tilde{c}_t) - u_t\rangle &\le \|\nabla_t\|_2 \|u(\tilde{c}_t) - u_t\|_2 \le \sqrt{6\eta m} \\
\sum_{t=1}^{T} \langle \nabla_t,\, u(\tilde{c}_t)\rangle \;-\; \sum_{t=1}^{T} \langle \nabla_t,\, u_t\rangle &\le \sqrt{6\eta m}\cdot T = \sqrt{6mT}. \qquad (70)
\end{aligned}$$

Now we bound the regret of playing the sampled c_t instead of $\tilde{c}_t$. We'll do this by showing that $\sum_{t=1}^{T} \frac{1}{m}\sum_{i=1}^{m} c_t(x_i) r_t(x_i, y_i)$ converges to $\sum_{t=1}^{T} \frac{1}{m}\sum_{i=1}^{m} \tilde{c}_t(x_i) r_t(x_i, y_i)$ using Azuma Hoeffding. Define the martingale difference sequence $Z_t = \frac{1}{m}\sum_{i=1}^{m} c_t(x_i) r_t(x_i, y_i) - \frac{1}{m}\sum_{i=1}^{m} \tilde{c}_t(x_i) r_t(x_i, y_i)$. This forms a bounded martingale difference sequence because the c_t is chosen only based on information from the past and $\tilde{c}_t = \mathbf{E}[c_t]$. Thus, applying Azuma Hoeffding, we have that with probability $1 - \delta/2$,

$$\left| \sum_{t=1}^{T} \frac{1}{m} \sum_{i=1}^{m} c_t(x_i) r_t(x_i, y_i) - \sum_{t=1}^{T} \frac{1}{m} \sum_{i=1}^{m} \tilde{c}_t(x_i) r_t(x_i, y_i) \right| \le O\left(\sqrt{T \ln 1/\delta}\right) \qquad (71)$$

Let us define the empirical gain for any hypothesis $c \in C$ at time t:

$$\hat{G}_t(c) = \frac{1}{m} \sum_{i=1}^{m} c(x_i) r_t(x_i, y_i)$$

and let $\hat{c}^* = \operatorname{argmax}_{c \in C} \sum_{t=1}^{T} \hat{G}_t(c)$ and $u^* = u(\hat{c}^*)$. With probability at least $1 - \delta/2$, the following regret bound holds:

$$\begin{aligned}
\sum_{t=1}^{T} \hat{G}_t(c_t) &\ge \sum_{t=1}^{T} \hat{G}_t(\tilde{c}_t) \;-\; O\left(\sqrt{T \ln 1/\delta}\right) && \text{by (71)} \\
&= -\frac{1}{m} \sum_{t=1}^{T} \langle \nabla_t,\, u(\tilde{c}_t)\rangle \;-\; O\left(\sqrt{T \ln 1/\delta}\right) \\
&\ge -\frac{1}{m} \sum_{t=1}^{T} \langle \nabla_t,\, u_t\rangle \;-\; O\left(\sqrt{T/m} \;+\; \sqrt{T \ln 1/\delta}\right) && \text{by (70)} \\
&\ge -\frac{1}{m} \sum_{t=1}^{T} \langle \nabla_t,\, u^*\rangle \;-\; O\left(\sqrt{T} \;+\; \sqrt{T/m} \;+\; \sqrt{T \ln 1/\delta}\right) && \text{for any } u^* \in \mathcal{K}, \text{ by (69)} \\
&\ge -\frac{1}{m} \sum_{t=1}^{T} \langle \nabla_t,\, u^*\rangle \;-\; O\left(\sqrt{T \ln 1/\delta}\right) && \text{simplifying } O(\cdot) \text{ expression} \\
&= \sum_{t=1}^{T} \hat{G}_t(\hat{c}^*) \;-\; O\left(\sqrt{T} \ln 1/\delta\right).
\end{aligned}$$

Now consider the population gain:

$$G_t(c) = \mathop{\mathbf{E}}_{(x,y)\sim\mathcal{D}}[c(x)r_t(x,y)]$$

The population regret of playing the sampled c_t can be decomposed as:

$$\max_{c\in\mathcal{C}} \sum_{t=1}^{T} G_t(c) - \sum_{t=1}^{T} G_t(c_t) \le \underbrace{\max_{c\in\mathcal{C}} \sum_{t=1}^{T} \hat{G}_t(c) - \sum_{t=1}^{T} \hat{G}_t(c_t)}_{\text{FTRL regret}} + \underbrace{\sum_{t=1}^{T} (\hat{G}_t(c_t) - G_t(c_t)) + \sum_{t=1}^{T} (G_t(c^*) - \hat{G}_t(c^*))}_{\text{Generalization error}},$$

where $c^* = \text{argmax}_{c\in\mathcal{C}} \sum_{t=1}^{T} G_t(c)$.

Using standard uniform convergence bounds based on Rademacher complexity, we have that with probability $1 - \delta/2$, for any $c \in \mathcal{C}, r \in \mathcal{R}$:

$$\left| \frac{1}{m} \sum_{i=1}^{m} c(x_i)r(x_i,y_i) - \mathop{\mathbf{E}}_{(x,y)\sim\mathcal{D}}[c(x)r(x,y)] \right| \le O\left(\sqrt{\frac{\ln 1/\delta}{m}} \right) + O(\mathsf{rad}_m(\mathcal{C}\cdot\mathcal{R})) \tag{72}$$

Therefore,

$$\left| (G_t(c) - \hat{G}_t(c)) \right| \le O\left(\sqrt{\frac{\ln 1/\delta}{m}} \right) + O(\mathsf{rad}_m(\mathcal{C}\cdot\mathcal{R})),$$

where $\mathsf{rad}_m(\mathcal{C}\cdot\mathcal{R})$ is the Rademacher complexity of the product class $\mathcal{C}\cdot\mathcal{R}$.

Combining the bounds for the regret bounds and the generalization error, we have that with probability at least $1 - \delta$:

$$\max_{c\in\mathcal{C}} \sum_{t=1}^{T} G_t(c) - \sum_{t=1}^{T} G_t(c_t) \le O\left(\sqrt{T \ln 1/\delta} \right) + O(T \cdot \mathsf{rad}_m(\mathcal{C}\cdot\mathcal{R})).$$

□

Lemma 7.5 (Hazan (2023))**.** Given a closed, convex decision set $\mathcal{K}$ and a strongly convex, smooth and twice differentiable regularizer $R : \mathcal{K} \to \mathbb{R}$. For every $u \in \mathcal{K}$, the Follow the Regularized Leader algorithm attains the following regret bound

$$\text{Reg}(T) \le 2\eta \sum_{t=1}^{T} \|\nabla_t\|_t^{*2} + \frac{R(u) - R(u_1)}{\eta}$$

where $u_t \in \mathcal{K}$ is the action at timestep t, ∇_t is the gradient at u_t of the loss function at time t, and $\|\cdot\|_t^* = \|\cdot\|_{\nabla^{-2}R(u_t)}$ represents the dual norm defined by the regularizer R at u_t.

7.2 Frank-Wolfe reduction to ERM oracle

In this section we explain how to implement a regularized ERM oracle over $\mathcal{C}$ by a sequence of calls to a standard ERM oracle over $\mathcal{C}$. The algorithm for the regularized ERM oracle is presented below as Algorithm 7. Its analysis is summarized by the following lemma.

Lemma 7.6 (Frank-Wolfe with explicit convex combination)**.** Given a dataset $\{(x_i, y_i)\}_{i=1}^m$, a class of functions $C \subseteq \{c : \mathcal{X} \to [-1, 1]\}$, a ERM oracle for C, and parameters $\eta, \epsilon > 0$, Algorithm 7 returns an ϵ-approximate solution c^* to the entropy-regularized ERM problem

$$\arg\min_{c \in C}^{\varepsilon} \left\{ -\eta \sum_{i=1}^{m} c(x_i) y_i + \sum_{i=1}^{m} (c(x_i) + 2) \log(c(x_i) + 2) \right\}$$

after $O(m/\epsilon)$ iterations. Moreover, the returned c^* is a convex combination of functions in C, i.e., $c^* = \sum_j \alpha_j c_j$ with $\sum_j \alpha_j = 1$ and $\alpha_j \geq 0$, and the algorithm also provides the weights $\{\alpha_j\}$.

Overview of Algorithm 7: Each iteration of Algorithm 6 requires (approximately) solving a constrained concave minimization problem $\min\{g(u) \mid u \in \mathcal{K}\}$ where

$$g(u) = -\eta \sum_{s=1}^{t} \sum_{i=1}^{m} r_s(x_i, y_i) u_i \; + \; \sum_{i=1}^{m} (u_i + 2) \log(u_i + 2) \tag{73}$$

and, as in the proof of (7.4), $\mathcal{K}$ denotes the convex hull of the set of vectors $u(c) = (c(x_1), c(x_2), \ldots, c(x_m))$ as c ranges over C. In this section we assume we are given a *ERM oracle* for C, that is, an algorithm for selecting the $c \in C$ that minimizes $\sum_{i=1}^m w_i c(x_i)$ for a given set of (x_i, w_i) pairs. This is equivalent to an oracle for minimizing linear functions over $\mathcal{K}$. Algorithm 7 below uses the ERM oracle to implement the Frank-Wolfe method, also known as conditional gradient descent, for approximately minimizing the convex function $g(u)$ over $\mathcal{K}$. At each iteration, the algorithm computes the gradient of the objective function, invokes the ERM oracle to find an extreme point in the original function class C, and then updates the current solution by forming a convex combination. After $O(\frac{m}{\varepsilon})$ iterations, it returns an ε-approximate solution to the original problem, expressed as a convex combination of functions from C, together with the associated weights. Since Algorithm 6 calls Algorithm 7 using the parameters $m = T$, $\varepsilon = \eta = \sqrt{1/T}$, we see that each such call requires $O(T^{3/2})$ iterations, hence $O(T^{3/2})$ calls to the ERM oracle.

Lemma 7.7 (Conditional Gradient Descent; (Hazan, 2023))**.** Let $K \subset \mathbb{R}^n$ with bounded ℓ_2 diameter R. Let f be a β-smooth function on K, then the sequence of points $x_t \in K$ computed by the conditional gradient descent algorithm satisfies

$$f(x_t) - f(x^*) \leq \frac{2\beta R^2}{t + 1}$$

for all $t \geq 2$ where $x^* \in \arg\min_{x \in K} f(x)$

Proof of Lemma 7.6. The objective function $g : \mathcal{K} \to \mathbb{R}$ defined in Equation (73) is well-defined and differentiable on $\mathcal{K}$. Its gradient is

$$\nabla g(u) = \begin{bmatrix} -\eta z_1 + \log(u_1 + 2) + 1 \\ -\eta z_2 + \log(u_2 + 2) + 1 \\ \vdots \\ -\eta z_m + \log(u_m + 2) + 1 \end{bmatrix}$$

where z_i denotes the sum $z_i = \sum_{s=1}^t r_s(x_i, y_i)$.

Algorithm 7 Frank-Wolfe for Entropy Regularized ERM with Explicit Convex Combination

procedure FrankWolfe($\{(x_i, y_i)\}_{i=1}^m, C, \eta, \varepsilon$)
  Initialize c_1 to an arbitrary function in C
  Let $\alpha_1 = 1$ and let $\alpha_j = 0$ for all $j > 1$
  for $t = 1, 2, \ldots, T$ **do**
    For each $i \in [m]$, set $w_i = -\eta y_i + \log(c_t(x_i) + 2) + 1$
    Call the ERM oracle on $\{(x_i, w_i)\}_{i=1}^m$ to obtain $s_t \in C$
    Set $\gamma_t = \frac{2}{t+1}$
    Update $c_{t+1} = (1 - \gamma_t)c_t + \gamma_t s_t$
    Update the weights of the convex combination:

$$\text{For } j < t : \alpha_j \leftarrow (1 - \gamma_t)\alpha_j, \quad \alpha_t \leftarrow \alpha_t + \gamma_t$$

    Since c_1 was chosen initially, we now have a convex combination:

$$c_{t+1} = \sum_{j=1}^{t} \alpha_j s_j + \Big(1 - \sum_{j=1}^{t} \alpha_j\Big) c_1.$$

  (Initially, $\alpha_1 = 1$ and no s_j are chosen, so c_1 is the starting point. After each iteration, we redistribute weights accordingly. Note that the α_j for $j > 1$ were initially zero and only become nonzero when their corresponding s_j appears.)

  return $c_1, s_1, \ldots, s_T$ and the final weights $(\alpha_1, \alpha_2, \ldots, \alpha_T)$

Now we compute the smoothness parameter and the diameter of the domain. Over the domain $[-1, 1]$, the second derivative of $(u_i + 2)\log(u_i + 2)$ with respect to u_i is at most 1. Thus, g is β-smooth with $\beta = 2$. The ℓ_2-diameter of K is at most $R = 2\sqrt{m}$. Applying Lemma 7.7 to g on K:

$$g(u_T) - g(u^*) \le \frac{2\beta R^2}{T+1} \le \frac{16m}{T+1}.$$

Choosing $T \ge \frac{16m}{\varepsilon}$ leads to $g(u_T) - g(u^*) < \varepsilon$. □

References

Abernethy, J., Bartlett, P. L., and Hazan, E. (2011). Blackwell approachability and no-regret learning are equivalent. In Kakade, S. M. and von Luxburg, U., editors, *Proceedings of the 24th Annual Conference on Learning Theory*, volume 19 of *Proceedings of Machine Learning Research*, pages 27–46, Budapest, Hungary. PMLR. 2, 6, 18

Arora, S., Hazan, E., and Kale, S. (2012). The multiplicative weights update method: a meta-algorithm and applications. *Theory of computing*, 8(1):121–164. 27, 72

Bartlett, P. L. and Mendelson, S. (2003). Rademacher and gaussian complexities: risk bounds and structural results. *J. Mach. Learn. Res.*, 3(null):463–482. 52

Beygelzimer, A., Kale, S., and Luo, H. (2015). Optimal and adaptive algorithms for online boosting. In *International Conference on Machine Learning*, pages 2323–2331. PMLR. 7, 32

Blackwell, D. (1956). An analog of the minimax theorem for vector payoffs. *Pacific Journal of Mathematics*, 6(1):1 – 8. 2, 6, 17

Błasiok, J., Gopalan, P., Hu, L., and Nakkiran, P. (2023). A unifying theory of distance from calibration. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, pages 1727–1740. 67

Brukhim, N., Chen, X., Hazan, E., and Moran, S. (2020). Online agnostic boosting via regret minimization. *Advances in Neural Information Processing Systems*, 33:644–654. 7, 32

Burhanpurkar, M., Deng, Z., Dwork, C., and Zhang, L. (2021). Scaffolding sets. *arXiv preprint arXiv:2111.03135*. 7

Casacuberta, S., Dwork, C., and Vadhan, S. (2024). Complexity-theoretic implications of multicalibration. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*, pages 1071–1082. 14

Chen, S.-T., Lin, H.-T., and Lu, C.-J. (2012). An online boosting algorithm with theoretical justifications. *arXiv preprint arXiv:1206.6422*. 7, 32

Dagan, Y., Daskalakis, C., Fishelson, M., Golowich, N., Kleinberg, R., and Okoroafor, P. (2024). Breaking the $t^{2/3}$ barrier for sequential calibration. 2, 14

Dawid, A. P. (1985). Calibration-based empirical probability. *The Annals of Statistics*, 13(4):1251–1274. 2

Dvoretzky, A., Kiefer, J., and Wolfowitz, J. (1956). Asymptotic minimax character of the sample distribution function and of the classical multinomial estimator. *The Annals of Mathematical Statistics*, pages 642–669. 74

Dwork, C., Hays, C., Immorlica, N., Perdomo, J. C., and Tankala, P. (2024). From fairness to infinity: Outcome-indistinguishable (omni)prediction in evolving graphs. 15

Dwork, C., Kim, M. P., Reingold, O., Rothblum, G. N., and Yona, G. (2021). Outcome indistinguishability. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 1095–1108. 4, 15

Dwork, C., Kim, M. P., Reingold, O., Rothblum, G. N., and Yona, G. (2022). Beyond bernoulli: Generating random outcomes that cannot be distinguished from nature. In *International Conference on Algorithmic Learning Theory*, pages 342–380. PMLR. 15

Foster, D. P. and Hart, S. (2018). Smooth calibration, leaky forecasts, finite recall, and nash dynamics. *Games and Economic Behavior*, 109:271–293. 67

Foster, D. P. and Vohra, R. V. (1998). Asymptotic calibration. *Biometrika*, 85(2):379–390. 14

Freund, Y. and Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139. 14

Garg, S., Jung, C., Reingold, O., and Roth, A. (2024). Oracle efficient online multicalibration and omniprediction. In *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2725–2792. SIAM. 2, 14, 15

Globus-Harris, I., Gupta, V., Jung, C., Kearns, M., Morgenstern, J., and Roth, A. (2023a). Multicalibrated regression for downstream fairness. In *Proceedings of the 2023 AAAI/ACM Conference on AI, Ethics, and Society*, pages 259–286. 15

Globus-Harris, I., Harrison, D., Kearns, M., Roth, A., and Sorrell, J. (2023b). Multicalibration as boosting for regression. *arXiv preprint arXiv:2301.13767*. 15

Gopalan, P., Hu, L., Kim, M. P., Reingold, O., and Wieder, U. (2023). Loss minimization through the lens of outcome indistinguishability. 2, 4, 5, 8, 9, 14, 15, 20, 21, 22, 23, 51, 73, 74, 75

Gopalan, P., Kalai, A. T., Reingold, O., Sharan, V., and Wieder, U. (2022). Omnipredictors. In Braverman, M., editor, *13th Innovations in Theoretical Computer Science Conference, ITCS 2022, January 31 - February 3, 2022, Berkeley, CA, USA*, volume 215 of *LIPIcs*, pages 79:1–79:21. Schloss Dagstuhl - Leibniz-Zentrum für Informatik. 1, 9, 14, 15, 16

Gopalan, P., Kim, M., and Reingold, O. (2024a). Swap agnostic learning, or characterizing omniprediction via multicalibration. *Advances in Neural Information Processing Systems*, 36. 15

Gopalan, P., Okoroafor, P., Raghavendra, P., Shetty, A., and Singhal, M. (2024b). Omnipredictors for regression and the approximate rank of convex functions. 8, 15, 40

Hazan, E. (2023). Introduction to online convex optimization. 56, 57

Hébert-Johnson, U., Kim, M. P., Reingold, O., and Rothblum, G. (2018). Multicalibration: Calibration for the (computationally-identifiable) masses. In *International Conference on Machine Learning*, pages 1939–1948. PMLR. 1, 4, 14, 15, 73

Hu, L., Navon, I. R. L., Reingold, O., and Yang, C. (2023). Omnipredictors for constrained optimization. In *International Conference on Machine Learning*, pages 13497–13527. PMLR. 15

Hu, L., Tian, K., and Yang, C. (2024). Omnipredicting single-index models with multi-index models. 15

Hu, L. and Wu, Y. (2024). Calibration error for decision making. 2, 14, 62, 68

Kakade, S. M. and Foster, D. P. (2008). Deterministic calibration and nash equilibrium. *Journal of Computer and System Sciences*, 74(1):115–130. 67

Kakade, S. M., Kanade, V., Shamir, O., and Kalai, A. (2011). Efficient learning of generalized linear and single index models with isotonic regression. *Advances in Neural Information Processing Systems*, 24. 15

Kalai, A. T. and Sastry, R. (2009). The isotron algorithm: High-dimensional isotonic regression. In *COLT*, volume 1, page 9. 15

Kim, M. P., Ghorbani, A., and Zou, J. (2019). Multiaccuracy: Black-box post-processing for fairness in classification. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pages 247–254. 2, 4, 15, 73

Kim, M. P. and Perdomo, J. C. (2023). Making decisions under outcome performativity. In *14th Innovations in Theoretical Computer Science Conference (ITCS 2023)*, volume 251, page 79. Schloss Dagstuhl–Leibniz-Zentrum für Informatik. 15

Kleinberg, B., Leme, R. P., Schneider, J., and Teng, Y. (2023). U-calibration: Forecasting for an unknown agent. In *The Thirty Sixth Annual Conference on Learning Theory*, pages 5143–5145. PMLR. 2, 5, 11, 14, 16, 20, 23, 65, 66

Lazaric, A. and Munos, R. (2009). Hybrid stochastic-adversarial on-line learning. In *COLT*. Citeseer. 10, 45

Li, Y., Hartline, J. D., Shan, L., and Wu, Y. (2022). Optimization of scoring rules. In *Proceedings of the 23rd ACM Conference on Economics and Computation*, pages 988–989. 5, 11, 14, 20, 23

Luo, H., Senapati, S., and Sharan, V. (2024). Optimal multiclass u-calibration error and beyond. 14

Pfisterer, F., Kern, C., Dandl, S., Sun, M., Kim, M. P., and Bischl, B. (2021). mcboost: Multi-calibration boosting for r. *Journal of Open Source Software*, 6(64):3453. 73

Qiao, M. and Valiant, G. (2021). Stronger calibration lower bounds via sidestepping. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 456–466. 2

Rakhlin, A., Sridharan, K., and Tewari, A. (2015). Online learning via sequential complexities. *J. Mach. Learn. Res.*, 16(1):155–186. 34, 52

Roth, A. and Shi, M. (2024). Forecasting for swap regret for all downstream agents. In *Proceedings of the 25th ACM Conference on Economics and Computation (EC 2024)*. 2

Shalev-Shwartz, S. and Ben-David, S. (2014). *Understanding machine learning: From theory to algorithms*. Cambridge university press. 47

Sridharan, K. (2010). Lecture notes on fat-shattering dimension, supervised learnability. 19

Wu, C., Heidari, M., Grama, A., and Szpankowski, W. (2022). Expected worst case regret via stochastic sequential covering. 10, 45, 76, 77

Zhao, S., Kim, M., Sahoo, R., Ma, T., and Ermon, S. (2021). Calibrating predictions to decisions: A novel approach to multi-class calibration. *Advances in Neural Information Processing Systems*, 34:22313–22324. 15

A Relating Proper Calibration to Existing Notions of Calibration

In this appendix, we document how Proper Calibration compares to prior notions of calibration. The standard notion of calibration, ℓ_1-calibration, being the strongest notion generally considered, implies proper calibration. While other notions previously considered do not imply proper calibration, proper calibration implies a number of existing notions such as U-calibration, smooth calibration, and ℓ_∞-calibration.

The one exception is the recent notion of Calibration Decision Loss (CDL) introduced by Hu and Wu (2024) to guarantee swap regret simultaneously for decision-making. We show that CDL is incomparable to bounded proper calibration, and thus, incomparable to Decision OI. In particular, CDL cannot be used in the existing Loss OI framework to guarantee omniprediction, but it also need not be satisfied by our omnipredictors.

We will simplify notation by dropping the contexts x_t, that is, we will only consider predictions $p_t = p_t(x_t)$ and outcomes y_t.

A.1 Calibration is stronger than Proper Calibration

Recall, one way to define ℓ_1-calibration is as weighted calibration according to *all* weight functions.

Definition (ℓ_1-calibration)**.** For a sequence of T predictions $\mathbf{p}$ and outcomes $\mathbf{y}$,

$$\ell_1\text{-CalErr}(\mathbf{p}, \mathbf{y}) = \left| \sup_{w:[0,1]\to[-1,1]} \sum_{t=1}^{T} w(p_t)(y_t - p_t) \right|.$$

From this definition, we immediately conclude that the ℓ_1-calibration error upper bounds the proper calibration error, as $\mathcal{W}_{\text{proper}} \subseteq \{w : [0,1] \to [-1,1]\}$ is a strict subset of all functions from $[0,1]$ to $[-1,1]$.

Proper Calibration does not imply ℓ_1-calibration. Here we argue that proper calibration is weaker than ℓ_1-calibration by exhibiting a sequence of predictions, where the proper calibration error is bounded by a constant but the ℓ_1-calibration error grows asymptotically with T.

Example A.1. Consider a sequence of T outcomes $\mathbf{y}$ that come in $m = 2k$ epochs of equal length, for an even k. For each $i \in \{1, 2, \ldots, 2k\}$, in the ith epoch the prediction for each time period $p_t = i/m$. In the first k epochs, where $i \in \{1, \ldots, k\}$, if i is even, the outcomes are all 0; if i is odd, then an i/k fraction are 1. In the second k epochs where $i \in \{k+1, \ldots, 2k\}$, if i is even, then a $(i-k)/k$ fraction are 1; if i is odd, the outcomes are all 1.

In this example, the predictions and outcomes are chosen such that the sign of the difference $y_i - p_i$ alternates in each epoch.

i	1	2	3	…	k	$k+1$	$k+2$	…	$2k-1$	$2k$
$\mathbf{p}$	$1/m$	$2/m$	$3/m$	…	k/m	$(k+1)/m$	$(k+2)/m$	…	$(2k-1)/m$	$2k/m$
$\mathbf{y}$	$1/k$	0	$3/k$	…	0	1	$2/k$	…	1	1
$y_i - p_i$	$+1/m$	$-2/m$	$+3/m$	…	$-1/2$	$1/2 - 1/m$	$-1/2 + 2/m$	…	$-1/m$	0

To analyze the calibration regrets for this sequence, consider the difference between predicted values in the ith epoch and the outcomes.

Upper bound on proper calibration regret. Consider the difference between the predicted values in the ith epoch and the outcomes. For the first k epochs, where $i \in \{1, \ldots, k\}$, the difference between $y_i - p_i$ in the ith epoch is given as:

$$\frac{(-1)^{i+1} \cdot i}{m}$$

For the second k epochs, where $i \in \{k+1, \ldots, 2k\}$, the difference between $y_t - p_t$ is given as:

$$\frac{(-1)^{2k-i+1} \cdot (2k-i)}{m} = \frac{(-1)^{j+1} \cdot j}{m}$$

for $j = 2k - i$, ranging from $j = k - 1$ down to 0.

The proper calibration error of this sequence can be bounded as follows.

$$\begin{aligned}
\sup_{s\in\{\pm\},\theta\in[0,1]} \sum_{t=1}^{T} s \cdot \mathrm{Th}_\theta(p_t) \cdot (y_t - p_t) &= \frac{T}{m} \cdot \sup_{s\in\{\pm\},\theta\in[0,1]} \sum_{i=1}^{m} s \cdot \mathrm{Th}_\theta(p_i) \cdot (y_i - p_i) \\
&= \frac{T}{m} \cdot \sup_{s\in\{\pm\},\theta\in[0,1]} s \cdot \left(\sum_{i=1}^{k} \mathrm{Th}_\theta(p_i) \cdot \frac{(-1)^{i+1} \cdot i}{m} + \sum_{j=0}^{k-1} \mathrm{Th}_\theta(p_{2k-j}) \frac{(-1)^{j+1} \cdot j}{m} \right) \\
&\leq \frac{2T}{m} \cdot \left| \sum_{i=0}^{k} \frac{(-1)^{i+1} \cdot i}{m} \right|
\end{aligned}$$

where the final inequality follows from the fact that the choice of threshold that maximizes the regret is in between $p_k = k/m = 1/2$ and $p_{k+1} = k + 1/m = 1/2 + 1/m$, with a negative sign $s = -1$.

The magnitude of $\left|\sum_{i=1}^{k} (-1)^{i+1} \cdot i\right|$ scales linearly with k, so in all, the regret is bounded as follows.

$$\frac{2T}{m} \cdot \left| \sum_{i=0}^{k} \frac{(-1)^{i+1} \cdot i}{m} \right| \leq \frac{2T}{m} \cdot \frac{O(k)}{m} = O(T/k)$$

Lower Bound on ℓ_1-calibration regret. To track the ℓ_1-calibration error, each term from the sums above contribute with their absolute value within the summation.

$$\begin{aligned}
\sup_{w:[0,1]\to[-1,1]} \sum_{t=1}^{T} w(p_t)(y_t - p_t) &= \sum_{t=1}^{T} |y_t - p_t| \\
&\geq \frac{2T}{m} \cdot \sum_{i=1}^{k-1} \left| \frac{i}{m} \right|
\end{aligned}$$

The magnitude of $\sum_{i=1}^{k-1} |i|$ scales quadratically with k, so in all, the regret is bounded as follows.

$$\frac{2T}{m} \cdot \sum_{i=1}^{k-1} \left| \frac{i}{m} \right| \geq \frac{2T}{m} \cdot \frac{\Omega(k^2)}{m} = \Omega(T)$$

Separation. In our construction, there are at least $2k^2$ time steps (as each epoch needs to reason about fractions of outcomes to precision $1/k$), so taking $k = \Theta(\sqrt{T})$, the proper calibration regret is upper bounded by $O(\sqrt{T})$, whereas the ℓ_1-calibration regret is lower bounded by $\Omega(T)$.

A.2 Proper calibration is stronger than other prior notions

Here, we show that proper calibration implies a number of prior notions of calibration, but is not implied by them. For each of ℓ_∞-calibration, U-calibration, and smooth calibration, we give examples that show upper bounding regret with respect to any of these notions is insufficient to give the same asymptotic upper bound for proper calibration regret.

ℓ_∞-calibration

To define ℓ_∞-calibration, we have to fix prediction level sets to be some discretization of the $[0, 1]$ interval.

Definition. Fix $m \in \mathbb{N}$ and consider predictions that live in multiples of $1/m$, $p \in \{0, 1/m, 2/m, \ldots, 1\}$. For such a sequence of T predictions $\mathbf{p}$ and outcomes $\mathbf{y}$

$$\ell_\infty\text{-CalErr}(\mathbf{p}, \mathbf{y}) = \max_{v \in \{0,1/m,\ldots,1\}} \left| \sum_{t=1}^{T} \mathbf{1}(p_t = v)(y_t - p_t) \right|$$

Proper Calibration implies ℓ_∞-calibration. Fix a discretization parameter m, a predictor that predicts values in $\{0, 1/m, 2/m, \ldots, 1\}$. First we note that for any $i \in [m]$

$$\mathbf{1}(p = i/m) = \frac{1}{2}\left(\mathrm{Th}_{i/m}(p) - \mathrm{Th}_{(i+1)/m}(p)\right)$$

Thus, the ℓ_∞ calibration error can be bounded as follows:

$$\begin{aligned}
&\max_{v \in \{0,1/m,2/m,\ldots,1\}} \left| \sum_{t=1}^{T} \mathbf{1}(p_t = v) \cdot (y_t - p_t) \right| \\
&= \frac{1}{2} \cdot \max_{p \in \{0,1/m,2/m,\ldots,1\}} \left| \sum_{t=1}^{T} \left(\mathrm{Th}_{i/m}(p_t) - \mathrm{Th}_{(i+1)/m}(p_t)\right) \cdot (y_t - p_t) \right| \\
&\leq \sup_{s \in \{\pm\}, \theta \in [0,1]} \sum_{t=1}^{T} s \cdot \mathrm{Th}_\theta(p_t) \cdot (y_t - p_t)
\end{aligned}$$

ℓ_∞-calibration does not imply Proper Calibration. The separation between ℓ_∞-calibration and proper calibration leverages the fact that predictions that are proper calibrated cannot be consistently biased in the same direction across prediction intervals, whereas ℓ_∞-calibration only cares about the maximum deviation over prediction intervals.

Example A.2. Consider a sequence of T outcomes $\mathbf{y}$ that comes in m epochs of equal length. In the ith epoch for each $i \in \{1, \ldots, m\}$, an i/m fraction of the outcomes are 1 and the remainder are 0. Consider a sequence of predictions $\mathbf{p}$ supported on m predictions $\{0, 1/m, 2/m, \ldots, (m-1)/m\}$, where for each time period in the ith epoch, the prediction $p_t = (i-1)/m$.

In this example, we can lower bound the proper calibration error by the bias of the predictor. Consider the threshold at 0, $\mathrm{Th}_0(p) = 1$, and consider the regret associated with this trivial threshold.

$$\begin{aligned}\sup_{s\in\{\pm\},\theta\in[0,1]} \sum_{t=1}^{T} s\cdot \mathrm{Th}_\theta(p_t)\cdot(y_t - p_t) &\geq \sum_{t=1}^{T}(y_t - p_t)\\ &= \frac{T}{m}\cdot\sum_{i=1}^{m}\left(\frac{i}{m} - \frac{i-1}{m}\right)\\ &= \frac{T}{m}\cdot m\cdot\frac{1}{m}\\ &= \frac{T}{m}\end{aligned}$$

In other words, the predictions are consistently biased, and proper calibration detects this bias, with regret T/m.

The ℓ_∞-calibration error, however, only detects the maximum deviation.

$$\begin{aligned}\max_{v\in\{0,1/m,\ldots,1\}}\left|\sum_{t=1}^{T}\mathbf{1}(p_t = v)(y_t - p_t)\right| &= \frac{T}{m}\cdot\max_{i\in[m]}\left(\frac{i}{m} - \frac{i-1}{m}\right)\\ &= \frac{T}{m}\cdot\left(\frac{i}{m} - \frac{i-1}{m}\right)\\ &= \frac{T}{m^2}\end{aligned}$$

In our construction, there are at least m^2 time steps (as each epoch needs to reason about fractions of outcomes to precision $1/m$), so taking $m = \Theta(\sqrt{T})$, the ℓ_∞-calibration error is upper bounded by $O(1)$, whereas the proper calibration error is lower bounded by $\Omega(\sqrt{T})$.

U-Calibration

Recall that U-Calibration measures calibration in terms of the worst-case proper loss regret.

Definition (Kleinberg et al. (2023))**.** For a sequence of T predictions $\mathbf{p}$ and outcome $\mathbf{y}$, let $p^* = \frac{1}{T}\sum_t y_t$; then,

$$\mathrm{UCal}(\mathbf{p},\mathbf{y}) = \sup_{\ell\in\mathcal{L}_{\mathrm{proper}}}\left[\sum_{t=1}^{T}\ell(p_t,y_t) - \ell(p^*,y_t)\right]$$

Proper Calibration implies U-Calibration. We bound the U-calibration error in terms of the proper calibration error for any sequence. To do this, first consider the following equality; fixing p_t, y_t, for any $p\in[0,1]$:

$$\begin{aligned}\mathop{\mathbf{E}}_{y\sim\mathrm{Ber}(p_t)}[\ell(p,y)] - \ell(p,y_t) &= (p_t\cdot\ell(p,1) + (1-p_t)\cdot\ell(p,0)) - (y_t\cdot\ell(p,1) + (1-y_t)\cdot\ell(p,0))\\ &= (p_t\cdot\Delta\ell(p) + \ell(p,0)) - (y_t\cdot\Delta\ell(p) + \ell(p,0))\\ &= (p_t - y_t)\cdot\Delta\ell(p)\end{aligned}$$

Then, consider expanding the U-calibration error as follows.

$$\begin{aligned}
\sup_{\ell \in \mathcal{L}_{\text{proper}}} \left[\sum_{t=1}^{T} \ell(p_t, y_t) - \ell(p^*, y_t) \right] &= \sup_{\ell \in \mathcal{L}_{\text{proper}}} \left[\sum_{t=1}^{T} \ell(p_t, y_t) - \mathop{\mathbf{E}}_{y \sim \text{Ber}(p_t)} [\ell(p_t, y)] + \mathop{\mathbf{E}}_{y \sim \text{Ber}(p_t)} [\ell(p_t, y)] - \ell(p^*, y_t) \right] \\
&\leq \sup_{\ell \in \mathcal{L}_{\text{proper}}} \left[\sum_{t=1}^{T} \ell(p_t, y_t) - \mathop{\mathbf{E}}_{y \sim \text{Ber}(p_t)} [\ell(p_t, y)] + \mathop{\mathbf{E}}_{y \sim \text{Ber}(p_t)} [\ell(p^*, y)] - \ell(p^*, y_t) \right] \\
&= \sup_{\ell \in \mathcal{L}_{\text{proper}}} \left[\sum_{t=1}^{T} (y_t - p_t) \Delta\ell(p_t) + (y_t - p_t) \Delta\ell(p^*) \right] \\
&\leq \sup_{\ell \in \mathcal{L}_{\text{proper}}} \left| \sum_{t=1}^{T} (y_t - p_t) \Delta\ell(p_t) \right| + \sup_{\ell \in \mathcal{L}_{\text{proper}}} \left| \sum_{t=1}^{T} (y_t - p_t) \Delta\ell(p^*) \right| \\
&\leq 2 \cdot \sup_{\ell \in \mathcal{L}_{\text{proper}}} \left| \sum_{t=1}^{T} (y_t - p_t) \Delta\ell(p_t) \right|
\end{aligned}$$

where the final inequality follows by the fact that for all $\ell \in \mathcal{L}_{\text{proper}}$, the weight function $\Delta\ell(p^*)$ is a constant across all t. This constant function is realizable by some choice of proper loss applied to p_t. Thus, the U-Calibration error is upper bounded by a constant factor of the proper calibration error.

U-Calibration does not imply Proper Calibration. We present a sequence of predictions and outcomes where the U-calibration regret is ≤ 0, but the proper calibration error grows linearly in T.

Example A.3. Consider a sequence of T predictions $\mathbf{y}$ where $y_t = 1$ if $t > T/2$ and 0 otherwise. Also a sequence of T predictions $\mathbf{p}$ where $p_t = 0.9$ if $t > T/2$, and 0.1 otherwise.

First we show that the sequence of predictions are U-calibrated. To do this, we use the upper bound in Kleinberg et al. (2023) that UCal $\leq$ 2VCal and bound VCal. For any $v \in [0, 1]$, we show that $\sum_{t=1}^{T} \ell_v(p_t, y_t) - \ell_v(0.5, y_t) < 0$ (since 0.5 is the best fixed prediction). Plugging in the definition of $\ell_v(p, y) = (y - v)\text{sgn}(v - p)$

$$\sum_{t=1}^{T} \ell_v(p_t, y_t) = \frac{T}{2}(-v)\text{sgn}(v - 0.1) + \frac{T}{2}(1 - v)\text{sgn}(v - 0.9)$$

Similarly, the second term simplifies to

$$\sum_{t=1}^{T} \ell_v(0.5, y_t) = \frac{T}{2}(-v)\text{sgn}(v - 0.5) + \frac{T}{2}(1 - v)\text{sgn}(v - 0.5)$$

Now, we consider four cases for v:

- $v \leq 0.1$: In this case, all the $\text{sgn}(v - p)$ terms have the same sign, so the difference between the two sums is 0.
- $v \in (0.1, 0.5)$: The second terms of both expressions cancel out, but the first term of the second expression becomes positive, making the overall difference negative.

- $v \in (0.5, 0.9]$: Similar to case 2, the second terms cancel out, and the first term of the second expression becomes positive, making the overall regret negative
- $v > 0.9$: Similar to case 1, all the $\text{sgn}(v - p)$ terms have the same sign, so the difference is 0.

In all cases, we have shown that $\sum_{t=1}^{T} \ell_v(p_t, y_t) - \ell_v(0.5, y_t) < 0$. Therefore, the predictor is U-calibrated.

Now, we show that the sequence of predictions has linear proper calibration error. To show this, it suffices to consider weight functions $\Delta\ell_v(p) = \text{sgn}(v - p)$ for $v = 0.5$. The weighted calibration error for this weight function is

$$\sum_{t=1}^{T} \Delta\ell_v(p_t)(y_t - p_t)$$

we can split the sum into two parts

$$\sum_{t=1}^{T} \text{sgn}(v - p_t)(y_t - p_t) = T/2(0 - 0.1) + T/2(-1)(1 - 0.9) = -0.1T$$

Taking the absolute value shows that the proper calibration error of the predictor is linear in T.

Smooth Calibration

Recall that Smooth Calibration is defined as a notion of weighted calibration using Lipschitz weight functions.

Definition (Kakade and Foster (2008); Foster and Hart (2018))**.** For a sequence of T predictions $\mathbf{p}$ and outcome $\mathbf{y}$,

$$\text{SmoothCal}(\mathbf{p}, \mathbf{y}) = \max_{w \in \mathcal{W}_{\text{Lip}}} \left| \sum_{t=1}^{T} w(p_t(x_t))(y_t - p_t(x_t)) \right|$$

where $\mathcal{W}_{\text{Lip}}$ is the set of 1-Lipschitz functions from $[0, 1]$ to $[-1, 1]$.

Proper Calibration implies Smooth Calibration. Proper calibration can be characterized in terms of signed threshold functions. This characterization shows that proper calibration guarantees weighted calibration with respect to weight functions defined by the difference of monotone functions, as signed thresholds form a basis. All Lipschitz functions can be expressed as a difference of monotone functions, so proper calibration implies smooth calibration.

Consequently, this implication further implies that proper calibration implies an upper bound on the lower distance to calibration (Błasiok et al., 2023).

Smooth Calibration does not imply Proper Calibration. This separation follows from the fact that proper calibration can encode highly non-Lipschitz tests, by the characterization in terms of threshold functions. The following example highlights this difference.

Example A.4. Consider a sequence of T predictions $\mathbf{y}$ where $y_t = 1$ if $t > T/2$ and 0 otherwise. Also a sequence of T predictions $\mathbf{p}$ where $p_t = 1/2 - \varepsilon$ if $t > T/2$, and $1/2 + \varepsilon$ otherwise.

To upper bound the smooth calibration error, we consider an arbitrary weight function $w \in \mathcal{W}_{\text{lip}}$ and evaluate its weighted calibration error

$$\sum_{t=1}^{T} w(p_t)(y_t - p_t) = \frac{T}{2} w(1/2 - \varepsilon)(1/2 + \varepsilon) + \frac{T}{2} w(1/2 + \varepsilon)(-1/2 - \varepsilon)$$

which simplifies to $\frac{T}{2}(w(1/2-\varepsilon) - w(1/2+\varepsilon))(1/2+\varepsilon)$. Since w is 1-lipschitz, the difference $(w(1/2-\varepsilon) - w(1/2+\varepsilon))$ is bounded by ε in magnitude. Thus, taking absolute values, we obtain that for all 1-lipschitz weight functions w $\sum_{t=1}^{T} w(p_t)(y_t - p_t) \leq \varepsilon T$

To lower bound the proper calibration error, it suffices to evaluate a single weight function $w(p) = \text{sgn}(v-p)$ for $v = 0.5$.

$$\sum_{t=1}^{T} \text{sgn}(0.5 - p_t)(y_t - p_t) = T/2(1/2 + \varepsilon) + T/2(-1)(-1/2 - \varepsilon) \geq T/4$$

Thus, we can make the smooth calibration error arbitrarily small by setting ε appropriately, however, the proper calibration error will always remain linear in T.

A.3 Calibration Decision Loss is incomparable to Proper Calibration

Hu and Wu (2024) introduced the notion of Calibration Decision Loss (CDL), which is a notion of swap regret simultaneous for decision-making losses. For a sequence of predictions $\mathbf{p}$ and outcomes $\mathbf{y}$, and a time step t where $p_t = i$, define the empirical swap prediction as follows.

$$\hat{p}_t = \frac{1}{|\{s \in [T] : p_s = i\}|} \sum_{s=1}^{T} y_s \cdot \mathbf{1}[p_s = i]$$

CDL has a number of equivalent formulations (up to constant factors), but one is based on Bregman divergences between the selected predictions and the empirical swap predictions. The CDL takes a worst-case choice over Bregman divergences derived from proper losses, and specifically, the V-Shaped losses.

$$\text{CDL}(\mathbf{p}, \mathbf{y}) = \sup_{v \in [0,1]} \sum_{t=1}^{T} B_v(p_t, \hat{p}_t)$$

where $B_v(p, q) = u_v(q) - u_v(p) + \nabla u_v(p) \cdot (p - q)$ is a Bregman divergence defined by the potential function $u_v(p) = -\mathbf{E}_{y \sim \text{Ber}(p)}[\ell_v(y, p)]$ defined as the Bayes risk for the (negative) loss of the V-shaped loss ℓ_v.[6]

Some consequences of this definition that make comparison to proper calibration possible include the following.

- $-\ell_v(y, p) = (y - v) \cdot \text{sgn}(p - v)$
- $-\Delta\ell_v(p) = \text{sgn}(p - v)$
- $u_v(p) = -p \cdot \Delta\ell_v(p) - \ell_v(0, p) = (p - v) \cdot \text{sgn}(p - v)$

[6]Note to better compare to proper calibration, we formulate these quantities in terms of our parameterization of the V-shaped losses, whereas Hu and Wu (2024) formulate the divergence in terms of proper scoring functions. Our formulation is equivalent to theirs up to a small constant factor.

- $\nabla u_v(p) = -\Delta \ell_v(p) = \mathrm{sgn}(p - v)$

With these facts in place, we can expand the Bregman divergence $B_v(p, q)$ as follows.

$$\begin{aligned} B_v(p,q) &= u_v(q) - u_v(p) + \nabla u_v(p) \cdot (p - q) \\ &= (q - v) \cdot \mathrm{sgn}(q - v) - (p - v) \cdot \mathrm{sgn}(p - v) + \mathrm{sgn}(p - v) \cdot (p - q) \\ &= (\mathrm{sgn}(q - v) - \mathrm{sgn}(p - v)) \cdot (q - v) \end{aligned}$$

That is, the Bregman divergence is equal to 0 when q and p are on the same side of v, and otherwise is equal to $2 \cdot |q - v|$.

Calibration Decision Loss does not imply Proper Calibration. This separation follows from the fact that CDL does not detect consistent bias. In particular, Example A.2 gives a separation. As we showed earlier, in this example, the proper calibration regret is lower bounded by $\Omega(\sqrt{T})$. Thus, we need only to upper bound the CDL to show a separation.

Note that every prediction p_t is close to $\hat{p}_t$; that is, $|p_t - \hat{p}_t| \leq 1/m$. Further, the prediction intervals the empirical swap intervals do not cross. Formally, consider two epochs $i < j$; for all s in the ith epoch and all t in the jth epoch,

$$p_s < \hat{p}_s \leq p_t < \hat{p}_t.$$

Thus, for any threshold $\theta \in [0, 1]$, there is at most one epoch i such that the prediction $(i-1)/m$ and empirical swap prediction i/m are on opposite sides of θ. Thus, the CDL is upper bounded by the contribution from this epoch—matching the ℓ_∞-calibration regret.

$$\begin{aligned} \mathrm{CDL}(\mathbf{p}, \mathbf{y}) &= \sup_{\theta \in [0,1]} \sum_{t=1}^{T} B_\theta(p_t, \hat{p}_t) \\ &\leq \frac{T}{m} \cdot \max_{i \in [m],\ \varepsilon \in (0, 1/m)} B_{i/m - \varepsilon}\left(\frac{i-1}{m}, \frac{i}{m}\right) \\ &\leq \frac{T}{m} \cdot \frac{1}{m} \\ &= \frac{T}{m^2} \end{aligned}$$

Again, taking $m = \Theta(\sqrt{T})$, the Calibration Decision Loss is upper bounded by $O(1)$, but the proper calibration regret scales as $\Omega(\sqrt{T})$.

Proper Calibration does not imply Calibration Decision Loss. In fact, Example A.1 shows a separation between proper calibration error and calibration decision loss. As before, taking $k = \Theta(\sqrt{T})$, the proper calibration regret is upper bounded by $O(\sqrt{T})$. Thus, it remains to lower bound the CDL.

Consider choosing the Bregman divergence B_v for $v = 1/2$. When v sits between the predicted value and the empirical swap prediction, then by the definition of B_v, we get a contribution to the CDL. In particular,

in the first k epochs, the odd epochs from $i = k/2 + 1$ to $k - 1$ will be split and contribute to the CDL as:

$$\begin{aligned} 2 \cdot |\hat{p}_t - 1/2| &= 2 \cdot \left| \frac{i}{k} - \frac{1}{2} \right| \\ &= \frac{2i - k}{k} \\ &= \frac{2j}{k} \end{aligned}$$

for $j = i - k/2$ from $j = 1$ to $k/2 - 1$. In the second k epochs, the even epochs from $i = k + 2$ to $3k/2$ will be split and contribute to the CDL as:

$$\begin{aligned} 2 \cdot |\hat{p}_t - 1/2| &= 2 \cdot \left| \frac{(i-k)}{k} - \frac{1}{2} \right| \\ &= \frac{|2(i-k) - k|}{k} \\ &= \frac{|2i - 3k|}{k} \\ &= \frac{2j}{k} \end{aligned}$$

for $j = 3k/2 - i$ from $j = 0$ to $k/2 - 2$.

In combination, the CDL can be expressed as follows.

$$\begin{aligned} \sup_{v \in [0,1]} \sum_{t=1}^{T} B_v(p_t, \hat{p}_t) &\geq \frac{T}{m} \cdot \left(\sum_{\substack{j \text{: odd} \\ 0 < j < k/2}} \frac{2j}{k} + \sum_{\substack{j \text{: even} \\ 0 \leq j < k/2}} \frac{2j}{k} \right) \\ &= \frac{T}{m} \cdot \sum_{i=0}^{k/2-1} \frac{2j}{k} \end{aligned}$$

The summation $\sum_{i=0}^{k/2-1} j$ scales quadratically in k (with $m = 2k$), so overall, the regret is lower bounded as follows.

$$\frac{T}{m} \cdot \sum_{i=0}^{k/2} \frac{2j}{k} \geq \frac{T}{m} \cdot \frac{\Omega(k^2)}{k} = \Omega(T)$$

Thus, an upper bound on the proper calibration regret does not imply the same bound on the CDL.

B Online Multiaccuracy

In this section, we introduce online multiaccurary as an important primitive for our algorithms for online omniprediction, as well as future applications. For completeness, we present algorithms for achieving online multiaccuracy for finite hypothesis classes and for infinite hypothesis classes via online weak agnostic learning oracles.

B.1 Multiaccuracy for Finite Hypothesis Classes

Theorem B.1 (Multiaccuracy)**.** Given a finite class of hypothesis functions $\mathcal{H}$, Algorithm 8 guarantees expected $\mathcal{H}$ multiaccuracy error of $O\left(\sqrt{T \ln |\mathcal{H}|}\right)$

Overview of Algorithm 8: The algorithm is based on Blackwell's Approachability Theorem. We define a two player game where the adversary player selects $z_t = (x_t, y_t) \in \mathcal{X} \times \{0, 1\}$ and the learner selects $p_t : \mathcal{X} \to [0, 1]$. Both players are allowed to play randomized strategies but since the learner observes x_t, we can simplify things and only consider $y_t \in \{0, 1\}$ and $p_t = p_t(x_t)$. We design the payoff vector of this game to reflect our objective of $\mathcal{H}$ multiaccuracy respectively. Define

$$u_{h,s}(p_t, z_t) = s(y_t - p_t)h(x_t) \quad \text{for } h \in \mathcal{H}, s \in \{+, -\}$$

Observe that after T rounds of interaction, $\mathcal{H}$-MAErr$(\mathbf{p}, \mathbf{x}, \mathbf{y}) = \max_{h,s} \sum_{t\in[T]} u_{h,s}(p_t, z_t)$. Therefore, we design the learner's target set to be the set of all vectors u whose coordinates is bounded by 0.

We use exponential weights update method in Line 2 to generate sequence of halfspaces w^t with coordinates for every $h \in \mathcal{H}, s \in \{+, -\}$. Given a halfspace w^t, the algorithm follows the strategy described from Line 4 to Line 7.

Algorithm 8 Multiaccuracy Algorithm

Input: Hypothesis class $\mathcal{H} \subseteq \{h : \mathcal{X} \to [0, 1]\}$
Input: Sequence of samples $\{(x_1, y_1), \ldots, (x_T, y_T)\}$
Output: Sequence of (randomized) predictors $p_1, \ldots, p_T$

1: **for** each $t \in [T]$ **do**
2: Let $w^t_{h,s} := \frac{\exp\left(\eta \sum_{i=1}^{t-1} u_{h,s}(p_i, z_i)\right)}{\sum_{h',s'} \exp\left(\eta \sum_{i=1}^{t-1} u_{h',s'}(p_i, z_i)\right)}$ for all $h \in \mathcal{H}, s \in \{+, -\}$
3: Compute

$$f(x_t) = \sum_{h,s} w^t_{h,s} \cdot s \cdot h(x_t)$$

4: **if** $f(x_t) \le 0$ **then**
5: Predict $p_t(x_t) = 0$
6: **else if** $f(x_t) > 0$ **then**
7: Predict $p_t(x_t) = 1$
8: Observe x_t, predict $p_t(x_t)$, and then observe y_t

Lemma B.2 (Halfspace Approachability)**.** Given a halfspace w, the strategy described in Line 4 to Line 7 outputs p_t such that $\langle w, u(p_t, z_t)\rangle \le 0$ for any choice of z_t

Proof of Lemma B.2. We consider the cases in the strategy separately:

- Case 1: If $f(x_t) \le 0$, predict $p_t(x_t) = 0$. Thus, for $h \in \mathcal{H}, s \in \{+, -\}$

$$w_{h,s}u_{h,s}(p_t, z_t) = s(y_t - p_t)h(x_t)w_{h,s} = sy_th(x_t)w^t_{h,s}$$

  Summing over values of h, s, we get

$$\langle w, u(p_t, z_t)\rangle = y_t f(x_t) \le 0 \quad \text{for any choice of } y \in \{0, 1\}$$

- Case 2: If $f(x_t) > 0$, predict $p_t(x_t) = 1$. Thus, for $h \in \mathcal{H}, s \in \{+, -\}$

$$w_{h,s}u_{h,s}(p_t, z_t) = s(y_t - p_t)h(x_t)w_{h,s} = sy_t(h(x_t))w_{h,s}$$

  Summing over values of h, s, we get

$$\langle w, u(p_t, z_t)\rangle = (y_t - 1)f(x_t) \le 0 \quad \text{for any choice of } y \in \{0, 1\}$$

□

Lemma B.3 (Exponential Weight Updates Arora et al. (2012))**.** The exponential weight updates in Line 2 provide a sequence of vectors w^t such that

$$\max_{w:\|w\|_1=1} \left\langle w, \sum_{t\in[T]} u(p_t, z_t) \right\rangle \leq \sum_{t=1}^{T} \langle w^t, u(p_t, z_t)\rangle + O\left(\sqrt{T \ln |\mathcal{H}|}\right)$$

Proof of Theorem B.1. We wish to bound the multiaccuracy error, that is, $\max_{h,s} \sum_{t\in[T]} u_{h,s}$ where the expectation is over the randomness in the sampling of p_t. Note that this is at most $\max_{w:\|w\|_1=1} \left\langle w, \sum_{t\in[T]} u(p_t, z_t)\right\rangle$.

$$\begin{aligned}
\max_{w:\|w\|_1=1} \left\langle w, \sum_{t\in[T]} u(p_t, z_t) \right\rangle &\leq \sum_{t=1}^{T} \langle w^t, u(p_t, z_t)\rangle + O\left(\sqrt{T \ln |\mathcal{H}|}\right) && \text{(by Lemma B.3)} \\
&= \sum_{t=1}^{T} \langle w^t, u(p_t, z_t)\rangle + O\left(\sqrt{T \ln |\mathcal{H}|}\right) && \text{(by linearity of expectation)} \\
&\leq \left(\sum_{t=1}^{T} 0\right) + O\left(\sqrt{T \ln |\mathcal{H}|}\right) && \text{(by Lemma B.2)} \\
&\leq O\left(\sqrt{T \ln |\mathcal{H}|}\right)
\end{aligned}$$

□

B.2 Achieving Online Multiaccuracy using Online Weak Agnostic Learner

Theorem B.4 (Multiaccuracy with Online Weak Agnostic Learner)**.** Given a *possibly infinite* class of hypothesis functions C, and an online weak agnostic learner for C with failure probability δ and regret bound $\text{OracleReg}_C^\delta(T)$, Algorithm 9 guarantees, with probability at least $1-\delta$, a C-multiaccuracy error bound of $\text{OracleReg}_C^\delta(T)$.

Algorithm 9 Multiaccuracy with an online weak agnostic learner

Input: Hypothesis class $C \subseteq \{c : \mathcal{X} \to [-1, 1]\}$
Input: Online Weak Agnostic Learner for $\{-1, +1\} \cdot C$, denoted $\mathcal{A}_{\mathsf{OWAL}}$
Input: Sequence of samples $\{(x_1, y_1), \ldots, (x_T, y_T)\}$
Output: Sequence of (randomized) predictors $p_1, \ldots, p_T$

1: Initialize OWAL, $q_1 \leftarrow \mathcal{A}_{\mathsf{OWAL}}$
2: **for** $t = 1$ to T **do**
3: Compute predictor $p_t(x) = \mathbb{I}[q_t(x) > 0]$
4: Observe x_t, Predict $p_t(x_t)$, Observe y_t
5: Obtain $q_{t+1} \leftarrow \mathcal{A}_{\mathsf{OWAL}}(x_t, y_t - p_t(x_t))$ ▷ execute one timestep of the OWAL

Lemma B.5 (Halfspace Approachability)**.** For every timestep $t \in [T]$, the predictor $p_t(x) = \mathbb{I}[q_t(x) > 0]$ constructed from the OWAL output q_t satisfies the following guarantee for all $x \in \mathcal{X}, y \in \{0, 1\}$:

$$q(x)(y - p_t(x)) \leq 0$$

Proof of Lemma B.5. We consider two cases based on the sign of $q_t(x)$:

- $q_t(x) > 0$. In this case $p_t(x) = 1$.
  - If $y = 1$, then $y - p_t(x) = 0$, so $q_t(x)\,(y - p_t(x)) = 0$.
  - If $y = 0$, then $y - p_t(x) = -1$, so $q_t(x)\,(y - p_t(x)) < 0$.

  In both subcases, $q_t(x)\,(y - p_t(x)) \leq 0$.
- $q_t(x) \leq 0$. In this case $p_t(x) = 0$.
  - If $y = 0$, then $y - p_t(x) = 0$, so $q_t(x)\,(y - p_t(x)) = 0$.
  - If $y = 1$, then $y - p_t(x) = 1$, so $q_t(x)\,(y - p_t(x)) \leq 0$.

  In both subcases, $q_t(x)\,(y - p_t(x)) \leq 0$.

Hence, for all x and all $y \in \{0, 1\}$, we have $q_t(x)\,(y - p_t(x)) \leq 0$ □

Proof of Theorem B.4. The online weak agnostic learner outputs sequence of predictors for $\{+1, -1\} \cdot C$ $q_1, \ldots, q_T$ such that with probability at least $1 - \delta$

$$\max_{c \in C, s \in \{+,-\}} \sum_{t=1}^{T} s \cdot c(x_t)(y_t - p_t(x_t)) \leq \sum_{t=1}^{T} q(x_t)(y_t - p_t(x_t)) + \text{OracleReg}_C^{\delta}(T) \qquad \text{(OWAL guarantee)}$$

$$\leq \sum_{t=1}^{T} 0 + \text{OracleReg}_C^{\delta}(T) \qquad \text{(by Lemma B.5)}$$

$$\leq \text{OracleReg}_C^{\delta}(T)$$

The claim follows by observing that $C\text{-MAErr}(\mathbf{p}, \mathbf{x}, \mathbf{y}) = \max_{c \in C, s \in \{+,-\}} \sum_{t=1}^{T} s \cdot c(x_t)(y_t - p_t(x_t))$ □

C Proper Calibrated Multiaccuracy Boosting

The work of Gopalan et al. (2023) established the efficiency of the Loss OI omniprediction framework via boosting for simultaneous (ℓ_1-)calibration and multiaccuracy, drawing on the original algorithms for multi-group fairness (Hébert-Johnson et al., 2018; Kim et al., 2019; Pfisterer et al., 2021). In this appendix, we show that proper calibration auditing can replace ℓ_1-calibration auditing within this framework with immediate efficiency gains. While the algorithm only achieves ε^{-4} dependence on the approximation parameter, it produces deterministic omnipredictors, which may be desirable in comparison to the near-optimal omnipredictors we learn via Algorithm 5.

Before presenting the novel variant of the algorithm, we account for the sample complexity achieved by the algorithm presented in (Gopalan et al., 2023).

Theorem C.1 (Sample Complexity bound in (Gopalan et al., 2023))**.** Given a class of loss functions $\mathcal{L}$, a hypothesis class $\mathcal{H}$ and a weak agnostic learning oracle for $C = (\Delta\mathcal{L} \circ \mathcal{H})$, there exists an algorithm that outputs a $(\mathcal{L}, \mathcal{H}, \varepsilon)$-omnipredictor with probability $1 - \delta$

$$O\left(\frac{d_C + \ln(1/\delta)}{\varepsilon^4} + \frac{\ln(1/\varepsilon)}{\varepsilon^{10}}\right)$$

where d_C represents the VC (or fat-shattering) dimension of the class C

As mentioned, the sample complexity improvement comes from designing an algorithm that achieves multiaccuracy and low proper calibration error instead of low ℓ_1-calibration error. Core to this improvement is an efficient test for proper calibration.

Testing Proper Calibration via CDF estimation. Unlike ℓ_1-calibration error, we can measure the proper calibration error of an arbitrary predictor p regardless of the fact that we have infinitely many test weight functions. The key idea is to estimate the CDF of the predictor $F(v) = \mathbf{E}_{(x,y)\sim\mathcal{D}}\mathbb{I}[p(x) \geq v]$. Because the weight functions are monotone functions, the CDF contains all the relevant information for estimating $\mathbf{E}_{(x,y)\sim\mathcal{D}}[w(p(x))]$. We can also express the threshold calibration error as

$$\max_{v\in[0,1]} \mathop{\mathbf{E}}_{(x,y)\sim\mathcal{D}}[(y - p(x))(2\mathbb{I}[p(x) \geq v] - 1)]$$

Lemma C.2 (Testing Proper Calibration via CDF estimation)**.** Given a predictor p, sample access to a distribution $\mathcal{D}$, and an error probability δ, there exists an algorithm that returns True with probability $1 - \delta$ if $\text{PCalErr}_{\mathcal{D}}(p) \leq \varepsilon$. The algorithm uses

$$O\left(\frac{\ln(1/\delta)}{\varepsilon^2}\right)$$

Proof of Lemma C.2. The algorithm first collects $\frac{\ln(1/\delta)}{\varepsilon^2}$ samples to estimate the CDF $F(v) = \mathbf{E}_{(x,y)\sim\mathcal{D}}[\mathbb{I}[p(x) \geq v]]$. Then the algorithm identifies the set Q of the $\delta/2$-quantiles of the CDF. Then the algorithm collects fresh samples $\frac{\ln(1/\delta)}{\varepsilon^2}$ and estimates $\mathbf{E}_{(x,y)\sim\mathcal{D}}[w_q(p(x))(y - p(x))]$ for each quantile $q \in Q$ from the samples where $w_q(p) = \text{sgn}(q - p)$. Then the algorithm returns True if any estimate is greater than δ and False otherwise.

Now we show that the algorithm is correct. By DKW Inequality (Dvoretzky et al., 1956), we know the empirical CDF is close in infinity distance to the true CDF i.e $\|\hat{F} - F\|_\infty \leq \delta/2$. This implies that the value $\mathbf{E}_{(x,y)\sim\mathcal{D}}[w_v(p(x))(y - p(x))]$ for any $v \in [0, 1]$ is $\delta/2$ close to $\mathbf{E}_{(x,y)\sim\mathcal{D}}[w_q(p(x))(y - p(x))]$ for some $q \in Q$. The rest of the claim follows by uniform convergence of the estimates for $q \in Q$. □

Testing Proper Calibration using a Weak Learner for One Dimensional Thresholds. As defined in the first section, proper calibration can be equivalently captured by weighted calibration over $\{-1, 1\}-$threshold functions. In the next lemma, we show how to test if a function is proper-calibrated using a weak agnostic learner for one dimensional threshold functions.

Lemma C.3 (Testing Proper Calibration via Weak Learning)**.** Given a predictor p, sample access to a distribution $\mathcal{D}$, and an error probability δ, there exists an algorithm uses a weak agnostic learner for $\mathcal{W}_{\text{Th}} : \{q \to \text{sgn}(v - q) : v \in [0, 1]\}$ and returns True with probability $1 - \delta$ if $\text{PCalErr}_{\mathcal{D}}(p) \leq \varepsilon$. The algorithm uses

$$O\left(\frac{\ln(1/\delta)}{\varepsilon^2}\right)$$

Proof of Lemma C.3. We first describe the algorithm. The algorithm simply collects $m = O\left(\frac{\ln(1/\delta)}{\varepsilon^2}\right)$ samples from $\mathcal{D}$. Then the algorithm feeds the weak agnostic learner pairs $(u_i, v_i)_{i=1}^m$ where $u_i = p(x_i)$ and $v_i = (y_i - p(x_i))$ and returns True if the weak learner does not return $\perp$, False otherwise. Now we prove the correctness of the algorithm. Recall that the VC dimension of one dimensional threshold functions is 1. Therefore a weak learner for 1d threshold class only needs $O\left(\frac{\ln(1/\delta)}{\varepsilon^2}\right)$ samples for this hypothesis class. If the weak learner does not return $\perp$, then there exists a weight function in $\mathcal{W}_{\text{Th}}$ such that $\mathbf{E}_{(x,y)\sim\mathcal{D}}[w(p(x))(y - p(x))] \geq \varepsilon/2$, then the proper calibration error is at least that amount. □

Overview of Algorithm 10. The algorithm follows the same boosting-style of (Gopalan et al., 2023). Starting with a constant predictor $q_0(x) = \frac{1}{2}$, it iteratively updates this predictor until it is both C-multiaccurate and proper calibrated (up to an additive error of ε). In each iteration t, the algorithm draws fresh samples from $\mathcal{D}$ and calls the weak agnostic learning oracle for C to check if q_t is multiaccurate. If such a c exists, then it updates q_t using c, then terminates the iteration. If not, then the algorithm proceeds to check if q_t is proper calibrated by calling the weak agnostic learning oracle for one dimensional thresholds. If there exists a threshold function w that correlates with $y - q_t(x)$ using $q_t(x)$ as input, then the algorithm updates q_t using w. If not, the algorithm terminates.

We improve the dependence on ε in this result significantly and obtain the following bound.

Theorem C.4 (Improved Sample Complexity Upper Bound)**.** Given a class of loss functions $\mathcal{L}$, a hypothesis class $\mathcal{H}$ and a weak agnostic learning oracle for $C = (\Delta\mathcal{L} \circ \mathcal{H})$, Algorithm 10 outputs a $(\mathcal{L}, \mathcal{H}, \varepsilon)$-omnipredictor with probability $1 - \delta$

$$O\left(\frac{d_C + \ln(1/\delta)}{\varepsilon^4}\right)$$

where d_C represents the VC (or fat-shattering) dimension of the class C (as determined by the WAL oracle).

Proof of Theorem C.4. First we show that when Algorithm 10 terminates, then q_T is an $(\mathcal{L}, \mathcal{H}, \varepsilon)$-omnipredictor with probability at least $1 - \delta$. This follows from the weak agnostic learning guarantee. For the algorithm to terminate, both weak agnostic learners must return $\perp$. Since the weak agnostic learner for C returns $\perp$, then every hypothesis in C has correlation less than ε with q_T. Since the weak agnostic learner for one dimensional thresholds returns $\perp$, the proper calibration error of q_T is at most ε.

Now we show that Algorithm 10 terminates after $8/\varepsilon^2$ iterations. We do this by using the expected square distance of q_t to the Bayes-optimal predictor q^* as a potential function. We show that after each iteration, the squared distance reduces by $O(\varepsilon^2)$. The change in potential from the t-th iteration to the $t+1$-iteration can be expressed as follows:

$$\mathbf{E}[(q^*(x) - q_t(x))^2] - \mathbf{E}[(q^*(x) - q_{t+1}(x))^2]$$

Since $q_{t+1} = q_t + \frac{\varepsilon}{2} \cdot f$ where f is either a $c \in C$ or $w \circ q_t$ for some $w \in \mathcal{W}_{\text{Th}}$, we have

$$\begin{aligned}
&\mathbf{E}[(q^*(x) - q_t(x))^2] - \mathbf{E}[(q^*(x) - q_t(x) + \tfrac{\varepsilon}{2} \cdot f(x))^2] \\
&= \varepsilon \cdot \mathbf{E}[(q^*(x) - q_t(x))f(x)] - \tfrac{\varepsilon^2}{4} \cdot \mathbf{E}[f(x)^2]
\end{aligned}$$

By the weak agnostic learning assumption, we know that $\mathbf{E}[(q^*(x) - q_t(x))f(x)] \geq \frac{\varepsilon}{2}$. We also know that $\mathbf{E}[f(x)^2] \leq 1$ since $f \in [-1, 1]$. Thus, the change in potential is at least $\frac{\varepsilon^2}{4}$. Since the expected squared distance at the start is less than 2, the total number of iterations T is at most $8/\varepsilon^2$.

Algorithm 10 PCal+MABoost

Input: Error parameters $\varepsilon, \delta \in [0,1]$, Sample access to $\mathcal{D}$
Oracle access to a Weak Agnostic learner for $C = (\Delta\mathcal{L} \circ \mathcal{H})$, denoted $\mathsf{WAL}_{C,\varepsilon}$
Oracle access to a Weak Agnostic learner for one dimensional thresholds, denoted $\mathsf{WAL}_{\mathrm{Th},\varepsilon}$
Output: Predictor q_T.

$t \leftarrow 0$
$q_0(\cdot) \leftarrow \frac{1}{2}$ ▷ the algorithm can be initialized with an arbitrary predictor
$ma \leftarrow$ false
$pc \leftarrow$ false
while $\neg ma$ or $\neg pc$ **do**
  $c_{t+1} \leftarrow \mathsf{WAL}_{C,\varepsilon}\left(\{x_i, y_i - q_t(x_i)\}_{i=1}^m\right)$ ▷ where $(x_i, y_i)_{i=1}^m$ are fresh samples from $\mathcal{D}$
  if $c_{t+1} = \perp$ **then**
    $ma \leftarrow$ true
  else
    $ma \leftarrow$ false
    $h_{t+1}(\cdot) \leftarrow q_t(\cdot) + \frac{\varepsilon}{2} \cdot c_{t+1}(\cdot)$.
    $q_{t+1}(\cdot) \leftarrow \Pi(h_{t+1}(\cdot))$ ▷ where Π projects h_{t+1} onto $[-1,1]$
    $t \leftarrow t+1$.
    break
  $w_{t+1} \leftarrow \mathsf{WAL}_{\mathrm{Th},\varepsilon}\left(\{q_t(x_i), y_i - q_t(x_i)\}_{i=1}^m\right)$ ▷ where $(x_i, y_i)_{i=1}^m$ are fresh samples from $\mathcal{D}$
  if $w_{t+1} = \perp$ **then**
    $pc \leftarrow$ true
  else
    $pc \leftarrow$ false
    $h_{t+1}(\cdot) \leftarrow q_t(\cdot) + \frac{\varepsilon}{2} \cdot w_{t+1}(q_t(\cdot))$.
    $q_{t+1}(\cdot) \leftarrow \Pi(h_{t+1}(\cdot))$ ▷ where Π projects h_{t+1} onto $[-1,1]$.
    $t \leftarrow t+1$.
return q_t.

The algorithm collects at most $\frac{\mathrm{VC}+\ln(1/\delta)}{\varepsilon^2}$ fresh samples in each iteration. Generalization from this number of samples is guaranteed by the semantics of the weak agnostic learning oracle. Since the number of iterations T is at most $8/\varepsilon^2$, the total sample complexity is bounded by

$$O\left(\frac{\mathrm{VC}(C) + \ln(1/\delta)}{\varepsilon^4}\right)$$

□

D Implementing an OWAL with Offline Sample Complexity

The goal of this section is to recapitulate some key definitions and results from Wu et al. (2022) and explain how they imply the existence of an online weak agnostic learner with regret that scales with the offline sample complexity.

Definition D.1 (Definition 2 of Wu et al. (2022))**.** We say a class G of functions $\mathcal{X} \to [0,1]$ is a *stochastic global sequential cover* of a class $C \subseteq [0,1]^{\mathcal{X}}$ with respect to the class $\mathcal{P}$ of joint distributions over

$x_1, \ldots, x_T$ at scale $\alpha > 0$ and confidence $\delta > 0$, if for all $\mu \in \mathcal{P}$, we have

$$\Pr_{x_1,\ldots,x_T \sim \mu} [\exists c \in C, \forall g \in G, \exists t \in [T] \text{ s.t. } |c(x_t) - g(x_t)| > \alpha] \leq \delta.$$

We define the minimal size of G to be the *stochastic global sequential covering number* of C.

Wu et al. (2022) show that bounds on the stochastic global sequential covering number imply bounds on the regret for arbitraty convex, lipschitz losses.

Lemma D.1 (Theorem 3 of Wu et al. (2022))**.** Let $\ell(\cdot, y)$ be convex, L-Lipschitz, and bounded by 1 for any $y \in [0, 1]$, and let C be a set of functions $\mathcal{X} \to [0, 1]$. Let G_α^δ be a stochastic global sequential covering of $\mathcal{H}$ at scale α and confidence δ with respect to a class of distributions $\mathcal{P}$. Then there exists an algorithm that outputs a sequence of predictions $\hat{y}_1, \ldots, \hat{y}_T$ such that

$$\sup_{\mu \in \mathcal{P}} \mathbb{E}_{x_1,\ldots,x_T \sim \mu} \left[\sup_{y_1,\ldots,y_T} \left(\sum_{t=1}^{T} \ell(\hat{y}_t, y_t) - \inf_{c \in C} \sum_{t=1}^{T} \ell(c(x_t), y_t) \right) \right] \leq \inf_{0 \leq \alpha \leq 1} \left\{ \alpha L T + \sqrt{\frac{T}{2} \log |G_\alpha^\delta| + 1} \right\}.$$

The theorem follows from running a multiplicative weights algorithm over the stochastic cover of the hypothesis class. Now we state how the size of the cover scales with the combinatorial dimensions of the class.

Lemma D.2 (Theorem 6 of Wu et al. (2022))**.** For any binary-valued class C with finite VC-dimension, there exists a global sequential covering set G of C with respect to the class of all i.i.d. distributions over $x_1, \ldots, x_T$ at scale $\alpha = 0$ and confidence δ such that for $T \geq e^9$, we have

$$\log |G| \leq 5 \operatorname{VC}(C) \log^2 T + \log T \log(1/\delta) + \log T.$$

Lemma D.3 (Theorem 17 of Wu et al. (2022))**.** Let C be a class of functions $\mathcal{X} \to [0, 1]$ with the α-fat shattering number $d(\alpha)$. Then there exists a stochastic global sequential covering set G of C with respect to the class of all i.i.d. distributions over $x_1, \ldots, x_T$ at scale α and confidence δ such that

$$\log |G| \leq 8 d(\alpha/32)(\log T \log(1/\alpha))^4 + \log T \log(\log T/\delta) + O(1),$$

where $O(1)$ hides absolute constants that are independent of α, T, and δ.

Note that these bounds cannot be improved up to log factors. See Wu et al. (2022) for more.

Proof of Corollary 6.3. The proof of the claim follows mainly from Theorem 3 of Wu et al. (2022), stated in Lemma D.1. We can apply this theorem since our loss function is convex and 1-Lipschitz. The algorithm runs a multiplicative weights algorithm over a set of experts indexed by a stochastic global sequential covering (Definition D.1). The statement of the claim requires the outputs of the algorithm to be in C. Given an arbitrary stochastic covering G of C at scale α, it is straightforward to construct a stochastic covering $G' \subset C$ at scale 2α. Finally, the regret guarantee in Lemma D.1 scales with the VC (or fat-shattering dimension) of the class C after plugging the bounds on the stochastic global sequential covering to the guarantees of Lemma D.1. To ensure that each output hypothesis is in C [7], at each timestep, the multiplicative weights algorithm samples c_t from a distribution over G' instead of predicting the weighted combination $\sum_i w_t(i) c_i$. This only adds a factor of $\tilde{O}\left(\sqrt{T \ln 1/\delta}\right)$ to the regret term due to an application of Azuma-Hoedding to show that $\sum_{t=1}^{T} c_t(x_t) \cdot y_t$ converges to $\sum_{t=1}^{T} \mathbf{E}[c_t(x_t)] \cdot y_t$.

□

[7] C may not be closed under convex combinations