Online Multivalid Learning: Means, Moments, and Prediction Intervals

Varun Gupta
University of Pennsylvania, Philadelphia, PA, USA

Christopher Jung
University of Pennsylvania, Philadelphia, PA, USA

Georgy Noarov
University of Pennsylvania, Philadelphia, PA, USA

Mallesh M. Pai
Rice University, Houston, TX, USA

Aaron Roth
University of Pennsylvania, Philadelphia, PA, USA

Abstract

We present a general, efficient technique for providing contextual predictions that are “multivalid” in various senses, against an online sequence of adversarially chosen examples \((x, y)\). This means that the resulting estimates correctly predict various statistics of the labels \(y\) not just marginally – as averaged over the sequence of examples – but also conditionally on \(x \in G\) for any \(G\) belonging to an arbitrary intersecting collection of groups \(G\).

We provide three instantiations of this framework. The first is mean prediction, which corresponds to an online algorithm satisfying the notion of multicalibration from [5]. The second is variance and higher moment prediction, which corresponds to an online algorithm satisfying the notion of mean-conditioned moment multicalibration from [6]. Finally, we define a new notion of prediction interval multivalidity, and give an algorithm for finding prediction intervals which satisfy it. Because our algorithms handle adversarially chosen examples, they can equally well be used to predict statistics of the residuals of arbitrary point prediction methods, giving rise to very general techniques for quantifying the uncertainty of predictions of black box algorithms, even in an online adversarial setting. When instantiated for prediction intervals, this solves a similar problem as conformal prediction, but in an adversarial environment and with multivalidity guarantees stronger than simple marginal coverage guarantees.

2012 ACM Subject Classification Theory of computation → Online learning theory

Keywords and phrases Uncertainty Estimation, Calibration, Online Learning

Digital Object Identifier 10.4230/LIPIcs.ITCS.2022.82


Funding Gupta, Jung, Noarov, and Roth are supported in part by NSF grants CCF-1763307 and CCF-1934876, and a grant from the Simons Foundation. Pai is supported in part by NSF grant CCF-1763349.

Acknowledgements We thank Aaditya Ramdas for helpful discussions about conformal prediction, as well as pointers to the literature. We thank Sergiu Hart, Dean Foster, Drew Fudenberg, and Rakesh Vohra for helpful discussions about calibration, as well as pointers to the literature. We also thank Ashish Rastogi for discussions about uncertainty estimation in practice.
Consider the problem of making predictions about the prognoses of patients with an infectious disease at the early stages of a pandemic. To be able to guide the allocation of medical interventions, we may want to predict, from each patient’s observable features $x$, things such as the expected severity of the disease $y$ in two days’ time. And since we will be using these predictions to allocate scarce resources, we will want to be able to quantify the uncertainty of our predictions: perhaps by providing estimates of the variance of outcomes, or perhaps by providing prediction intervals at a desired level of confidence.

This is an online problem because we must start making predictions before we have much data, and the predictions are needed immediately upon the arrival of a patient. It is also a problem in which the environment is rapidly changing: the distribution of patients changes as the disease spreads through different populations, and the conditional distribution on outcomes given features changes as we learn how to better treat the disease.

How can we approach this problem? The conformal prediction literature [10] aims to equip arbitrary regression and classification procedures for making point predictions with prediction intervals that contain the true label with (say) 95% probability. But for the application in our example, conformal prediction has two well-known shortcomings:

### Marginal Guarantees

Conformal prediction only promises marginal prediction intervals: in other words, it provides guarantees that (e.g.) 95% of the prediction intervals produced over a sequence of predictions cover their labels. But these guarantees are averages over what are typically large, heterogeneous populations, and therefore provide little guidance for making decisions about individuals. For example, it would be entirely consistent with the guarantee of a 95% marginal prediction interval $[\ell_t, u_t]$ if for individuals from some demographic group $G$ making up less than 5% of the population, their labels $y_t$ fall outside of $[\ell_t, u_t]$ 100% of the time.\(^1\)

One could run many parallel algorithms for different demographic groups $G_i$, but then there would be no clear way to interpret the many different predictions one would receive for an individual belonging to several demographic groups at once ($x \in G_i$ for multiple groups $G_i$); for example, prediction intervals corresponding to different demographic groups could be disjoint. To see that marginal guarantees on their own are extremely weak, consider a batch (distributional) setting in which labelled points are drawn from a fixed distribution $D$: $(x, y) \sim D$. Then we could provide valid 95% marginal prediction intervals by entirely ignoring the features and giving a fixed prediction interval of $[\ell, u]$ for every point, where $[\ell, u]$ is such that $\Pr_{(x, y) \sim D}[y \notin [\ell, u]] = 0.05$.

### Distributional Assumptions

The conformal prediction literature almost exclusively assumes that the data is drawn from an exchangeable distribution (for example, i.i.d. data satisfies this property), and does not offer any guarantees when the data can quickly change in unanticipated or adversarial ways.

In this paper we give techniques for dealing with both of these problems (and similar issues that arise for the problem of predicting label means and higher moments) by drawing on ideas from the literature on calibration [1, 2]. Calibration is similar to conformal prediction

---

\(^1\) Even more insidious reversals, albeit not in the context of conformal prediction, have been observed on real world data – see the Wikipedia entry for Simpson’s paradox (https://en.wikipedia.org/wiki/Simpson%27s_paradox) for several examples.
in that it aims to give point estimates in nonparametric settings that satisfy marginal rather than conditional guarantees (i.e. that agree with the true distribution as averaged over the data rather than conditioned on the features of a particular data point). But calibration is concerned with predicting label expectations, rather than giving prediction intervals. Informally speaking, calibrated predictions satisfy that when averaging over all days when the prediction was (approximately) \( p \), the realized labels average to (approximately) \( p \), for all \( p \). Note that in a distributional setting, if a learner truly was predicting the conditional label expectations conditional on features \( p_x = \mathbb{E}_{(x,y) \sim \mathcal{D}}[y|x] \), then the forecasts would be calibrated – but just as with marginal prediction intervals, calibration on its own is a very weak condition in a distributional setting. For example, a learner could achieve calibration simply by making a single, constant prediction of \( p = \mathbb{E}_{(x,y) \sim \mathcal{D}}[y] \) for every point, and so calibrated predictions need not convey much information. Thus, just like the conformal prediction literature, the calibration literature is primarily focused on the online prediction setting. But from early on, the calibration literature has focused on the adversarial setting in which no distributional assumptions need to be made at all [2, 3, 9].

Calibration also suffers from the weaknesses that come with marginal guarantees: namely that calibrated predictions may have little to do with the conditional label expectations for members of structured sub-populations. [5] proposed an elegant solution to this problem in the batch setting, when predicting expectations, which they termed “multicalibration”. Informally speaking, a guarantee of multicalibration is parametrized by a large collection of potentially intersecting subsets of the feature space \( \mathcal{G} \) (corresponding e.g. to demographic groups or other categories relevant for the prediction task at hand). Multicalibration asks for predictions that are calibrated not only over the full distribution \( \mathcal{P} \), but also simultaneously over each of the induced distributions obtained by conditioning on membership in a set \( G \in \mathcal{G} \). Moreover, [5] showed how to obtain multicalibrated estimators in the batch, distributional setting with sample complexity that depends only logarithmically on \(|\mathcal{G}|\). [6] showed how to extend the notion of (multi)calibration from expectations to variances and other higher moments – and derived algorithms for obtaining such estimates in the batch setting.

1.1 Our Results and Techniques

In this paper, we give a general method for obtaining different kinds of “multivalid” predictions in an online, adversarial setting. This includes mean estimates that satisfy the notion of mean multicalibration from [5], moment estimates that satisfy the notion of mean-conditioned moment multicalibration from [6], and prediction intervals which satisfy a new notion of multivalidity, defined in this paper. The latter asks for tight marginal prediction intervals, which are simultaneously valid over each demographic group \( G \in \mathcal{G} \). We give a formal definition in Section 2 (and review the definition of mean multicalibration), but informally, multivalidity for prediction intervals asks, given a target coverage probability \( 1 - \delta \), that for each group \( G \in \mathcal{G} \) there be roughly a \( (1 - \delta) \)-fraction of points \((x_t, y_t)\) with \( x_t \in G \) whose label is contained within the predicted interval \((y_t \in [\ell_t, u_t])\). In fact, we ask for the stronger calibration-like guarantee, that these marginal coverage guarantees hold even conditional on the prediction interval, which (among other things) rules out the trivial solution that predicts the full interval with probability \( 1 - \delta \) and an empty interval with probability \( \delta \).

Due to space limitations, below we only focus on means and prediction intervals and defer the discussion of multicalibrated mean-moment predictions to the full version of this paper. Because our algorithms handle adversarially selected examples, they can equally well be used to augment arbitrary point prediction procedures which give predictions \( f_t(x_t) = \tilde{y}_t \), independently of how they are trained: We can simply feed our algorithms for multivalid
predictions with the residuals $\hat{y}_t - y_t$. For example, we can get variance estimates or prediction intervals for the residuals to endow the predictions of $f_t$ with uncertainty estimates. Endowing point predictors with prediction intervals in this way provides an alternative to conformal prediction that gives stronger-than-marginal (multivalid) guarantees, under much weaker assumptions (adversarially chosen examples). In general, for each of our techniques, if we instantiate them with the trivial group structure (i.e. one group, containing all points), then we recover standard (or slightly stronger) marginal guarantees: i.e. simple calibrated predictions and simple marginal prediction intervals. But as we enrich our collection of sets $G$, our guarantees become correspondingly stronger.

The General Strategy

We derive our online algorithms using a general strategy that dates back to [3], who used it to give online algorithms for the problem of simple calibration in a setting without features. In our context, the general strategy proceeds as follows:

1. Define a surrogate loss function, such that if the surrogate loss is small at the end of $T$ rounds, then the learner’s predictions satisfy our chosen notion of multivalidity over the empirical distribution of the history of the interaction.

2. For each round $t$, argue that if the learner knew the adversary’s chosen distribution over labeled examples, she would be able to make a prediction guaranteeing a small expected increase in the surrogate loss function at that round. This step is often straightforward, because once we fix a known data distribution $D$, “true distributional quantities” (such as conditional label expectations, conditional label variances, and conditional label quantiles) generally satisfy our corresponding multivalidity desideratum by design.

3. Appeal to the minimax theorem to conclude that there must therefore exist a randomized prediction strategy for the learner that guarantees that the expected increase in the surrogate loss function is small for any choice of the adversary.

Recently, this general strategy has been distilled and generalized in the followup work [8], where it is in particular also shown to recover various classic no-regret results and Blackwell approachability guarantees.

On its own, carrying out this strategy for a particular notion of multivalidity only shows the existence of a multivalid algorithm in the presence of an adversary. To implement this strategy efficiently, we thus also have to demonstrate how to compute at each round the equilibrium strategy whose existence is shown in Step 3 above.

We instantiate this general strategy in Section 4 for the case of mean multicalibration, which also serves as a template for our derivation and analysis of algorithms for prediction interval multivalidity in Section 5. The framework of our analysis is the same in each case, but the details differ: to carry out Step 2, we must bound the value of a different game, and to carry out Step 3, we must solve for the equilibrium of a different game. In each case, we obtain efficient online algorithms for obtaining high probability $\alpha$-approximate multivalidity bounds (of different flavors), with $\alpha$ scaling roughly as $\alpha \approx \sqrt{\log |G|/T}$, over interactions of length $T$; see Sections 4.1 and 5.1 for exact theorem statements. In all cases, our algorithms

---

2 In fact, even with the trivial group structure, our guarantees (with appropriately set parameters) remain stronger than marginal coverage. This is because our prediction intervals remain valid even conditioning on the prediction that we made. For example, a prediction interval $(\ell, u)$ is valid not just as averaged over all rounds, but also as averaged over those rounds $t$ when we made that specific prediction: $(\ell_t, u_t) = (\ell, u)$.

3 See also the argument by Sergiu Hart, communicated in [2] and detailed in [4].
have per-round runtime that is linear in $|G|$, and polynomial in the other parameters of the problem. In fact, both our run-time and our convergence bounds can be improved if each individual appears in only a bounded number of groups. Our algorithms can at each step $t$ be implemented in time linear in the number of groups $G \in G$ that contain the current example $x_t$. This is linear in $|G|$ in the worst case, but can be substantially smaller. Similarly, we show in the full version of this paper that if each individual appears in at most $d$ groups, then the $\log |G|$ term in our convergence bounds can be replaced with $\log(d)$, which gives informative bounds even if $G$ is infinitely large. Without assumptions of this sort, runtime polynomial in $|G|$ (rather than logarithmic in $|G|$, as our convergence bounds are) is necessary in the worst case, even for mean multicalibration in the offline setting, as shown by [5].

Adapting the original approach of [3] runs into several obstacles, stemming from the fact that the action space of both the learner and the adversary and the number of constraints defining our calibration desideratum are both much larger in our setting. Consider the case of mean prediction – in which the goal is to obtain calibrated predictions. In the featureless setting studied by [3], the action space for the learner corresponds to a discretization of the real unit interval $[0, 1]$, and the action space of the adversary is binary. In our setting, in which data points are endowed with features from a large feature space $X$, the learner’s action space corresponds to the set of all functions mapping $X$ to $[0, 1]$, and the adversary’s action space corresponds to the set of all labelled examples $X \times [0, 1]$. Similarly, for simple calibration, the number of constraints is equal to the chosen discretization granularity of the unit interval $[0, 1]$, whereas in our case, the number of constraints also grows linearly with $|G|$, the number of groups over which we want to be able to promise guarantees.

Convergence Rates and Sample Complexity

The surrogate loss function used by [3] bounds the $\ell_2$ calibration error – i.e. the average squared violation of all of the constraints used to define calibration. Because all of the notions of multivalidity that we consider consist of a set of constraints of size scaling linearly with $|G|$, if we attempted to bound the $\ell_2$ violation of our multivalidity constraints, we would necessarily obtain convergence bounds that are polynomial in $|G|$. Instead we use a different surrogate loss function – a sign-symmetrized version of an exponential soft-max – that can be used to bound the $\ell_\infty$ violation of our multivalidity constraints, and allows us to obtain bounds that scale only logarithmically with $|G|$. We note that $\ell_\infty$ violation is consistent with how the existing literature on batch multicalibration [5] has quantified approximation guarantees. In fact, by using standard online-to-offline reductions, we are able to derive new, optimal sample complexity bounds for mean and moment multicalibration for the batch distributional setting that improve on the sample complexity bounds given in [5, 6]. When applied to the batch setting, our online algorithms take only a single pass through the data, and avoid issues related to adaptive data reuse that complicated previous batch algorithms.5

Computation of Equilibrium Strategies

To compute equilibria of the large action space games we define, we do not attempt to directly compute or represent the function that we use at each round $t$ to map features to labels. Instead, we represent this function implicitly by “lazily” solving a smaller equilibrium

---

4 See the full version of this paper.
5 However, unlike previous batch algorithms which make deterministic predictions, the batch algorithm that we obtain through this reduction makes randomized predictions.
computation problem only after we have observed the adversary’s choice of feature vector \( x \) (but before we have observed the label \( y \)) to compute a distribution over predictions. We show that this computation is tractable in each of our settings. In the case of mean multicalibration, we are able to analytically derive a simple algorithm for sampling from an (approximate) equilibrium strategy, presented in Section 4.2.

On the other hand, for prediction interval multivalidity, we show in Section 5.2 that the equilibrium computation problem can be cast as a linear program. Naively, this linear program has infinitely many constraints, but we show that it can ultimately be represented using finitely many constraints and has an efficient separation oracle, allowing to solve it in polynomial time via the Ellipsoid algorithm.

Advantages of Conformal Prediction

We have thus far emphasized the advantages that our techniques have over conformal prediction – but we also want to highlight the strengths of conformal prediction relative to our work, and directions for future improvement. Conformal prediction aims to obtain marginal coverage with respect to some (unknown) underlying distribution. Under the distributional assumption, it is able to obtain coverage (over the randomness of the distribution) at a rate of \( 1 - \delta + O(1/T) \) [7]. In contrast, in our setting, there is no underlying distribution. We therefore give guarantees on empirical coverage – i.e the fraction of realized labels that are covered by our predicted intervals. As a result, our coverage bounds necessarily have \( O(1/\sqrt{T}) \) error terms. We note that conformal prediction methods also obtain empirical coverage on the order of \( 1 - \delta \pm O(1/\sqrt{T}) \), as our methods do [7]. Conformal prediction methods naturally give one sided coverage error on the distribution (i.e. the coverage probability is always \( \geq 1 - \delta \)), whereas our empirical coverage error is two-sided. We note that there is a simple but inelegant way to use our techniques to obtain one sided coverage: run our algorithms with coverage parameter \( 1 - \delta' = 1 - \delta / 2 \), and predict trivial coverage intervals until our error bounds are \( \leq \delta / 2 \). Techniques from the conformal prediction literature also can be applied to very general label domains \( Y \), and can be used to produce very general kinds of prediction sets. In our paper, we restrict attention to real-valued labels \( Y = [0,1] \) and prediction intervals. We believe that there are no fundamental obstacles to generalizing our techniques to other label domains and prediction sets, making it an interesting direction for future work. Finally, the conformal prediction literature has developed a number of very simple, practical techniques. In this paper, we give polynomial time algorithms, of varying complexity. Our algorithm for mean multicalibration in Section 4 is very simple to implement, but our algorithm for multivalid interval prediction in Section 5 requires solving a linear program with a separation oracle. Thus, another important direction for future work is reducing the complexity of our techniques and doing empirical evaluations.

2 Preliminaries

2.1 Notation

We let \( \mathcal{X} \) denote a feature domain and \( Y = [0,1] \) a label domain. We write \( \mathcal{G} \subseteq 2^Y \) to denote a collection of subsets of \( \mathcal{X} \). Given any \( x \in \mathcal{X} \), we write \( \mathcal{G}(x) \) for the set of groups that contain \( x \), i.e. \( \mathcal{G}(x) = \{ G \in \mathcal{G} : x \in G \} \). For any positive integer \( T \), we write \( [T] = \{1, \ldots, T\} \). In general, we denote random variables with tildes (e.g. \( \tilde{X}, \tilde{Y} \)) to distinguish them from their realizations (denoted e.g. \( X, Y \)). Given a finite set \( A \), we write \( \Delta A \) for the probability simplex over the elements in \( A \).

\[ \text{82:6 Online Multivalid Learning: Means, Moments, and Prediction Intervals} \]
2.2 Online Prediction

Online (contextual) prediction proceeds in rounds indexed by \( t \in [T] \), for a given finite horizon \( T \). In each round \( t \), an interaction between a learner and an adversary proceeds as follows:

1. The adversary chooses a joint distribution over feature vectors \( x_t \in X \) and labels \( y_t \in Y \).
   The learner receives \( x_t \) (a realized feature vector), but no information about \( y_t \) is revealed.
2. The learner chooses a distribution over predictions \( p_t \in P \) from some domain \( P \).
3. The learner observes \( y_t \) (a realized label).

For an index \( s \in [T] \), we denote by \( \pi_s \) the transcript of the interaction in rounds \( 1 \leq t \leq s \):

\[
\pi_s = ( ( x_t, p_t, y_t ) )_{t=1}^s.
\]
We write \( \Pi^* \) as the domain of all transcripts. Formally, the adversary is modelled as a probabilistic mapping \( \text{Adv} : \Pi^* \rightarrow \Delta(X \times Y) \) from transcripts to distributions over labelled data points, and the learner is modeled as a mapping \( \text{Learn} : \Pi^* \rightarrow (X \rightarrow \Delta P) \) from transcripts to a probabilistic mapping from feature vectors to distributions over predictions. An adversary may be either unconstrained – free to play any point in \( \Delta(X \times Y) \) – or constrained to choose from some specified subset of \( \Delta(X \times Y) \). Fixing both a learner and an adversary induces a probability distribution over transcripts. Our goal is to derive particular learning algorithms, and to prove that various kinds of bounds hold either in expectation, or with high probability over the randomness of the transcript, in the worst case over transcript distributions, where we quantify over all possible adversaries.

Given a transcript \( \pi_T \), a group \( G \in \mathcal{G} \) and a set of rounds \( S \subseteq [T] \), we write

\[
G_S = \{ t \in S : x_t \in G \}.
\]
In words, this is the set of rounds in \( S \) in which the realized feature vectors in the transcript belonged to \( G \). When it is clear from context, we sometimes overload notation, and for a group \( G \in \mathcal{G} \), and a period \( s \leq T \), write \( G_s \) to denote the set of data points (indexed by their rounds) in a transcript \( \pi_s \) that are members of the group \( G \):

\[
G_s = \{ t \in [s] : x_t \in G \}.
\]

2.2.1 Types of Predictions, and Notions of Multivalidity

We consider three types of predictions in this paper: Mean predictions, pairs of mean and higher moment (e.g., variance) predictions, and prediction intervals. As mentioned before, mean-moment predictions are not considered here due to space limitations,\(^8\) so we focus here only on the mean prediction setting and the prediction intervals setting.

**Mean Predictions**

In this setting, the prediction domain will be the unit interval: \( \mathcal{P}_{\text{mean}} = [0,1] \). On day \( t \), the learner selects \( p_t \equiv \bar{p}_t \in \mathcal{P}_{\text{mean}} \), with the goal of predicting the conditional label expectation \( \mathbb{E}[y_t|x_t] \). For any subset of days \( S \subseteq [T] \), we denote the true label population mean conditional on \( t \in S \) and the average of our mean estimates over days \( t \in S \), respectively, as

\[
\mu(S) = \frac{1}{|S|} \sum_{t \in S} y_t, \quad \bar{\mu}(S) = \frac{1}{|S|} \sum_{t \in S} \bar{p}_t.
\]

\(^7\) We will consider several different kinds of predictions in this paper, and so are agnostic to the domain of the prediction for now – we use \( P \) as a generic domain.

\(^8\) See the full version of this paper.
We will want our predictions to satisfy large numbers of mean consistency constraints: that conditional label averages be approximately equal to conditional prediction averages over various sets $S$.

**Definition 1 (Mean Consistency).** Given a transcript $\pi_T$, we say that the mean predictions $\{\mu_t\}_{t=1}^T$ are $\alpha$-mean consistent\(^9\) on $S \subseteq [T]$ , if

$$|\mu(S) - \mu(S)| \leq \alpha \frac{T}{|S|}.$$ 

Before defining mean multicalibration, recall that informally, a sequence of mean predictions is calibrated if over the days when $p_t$ is $\approx p$, the average realized label $y_t$ is $\approx p$, for all $p$. To collect mean predictions $\mu_t$ that are approximately equal to $p$ for each $p$, we group real-valued predictions into $n$ buckets of width $\frac{1}{n}$. The parameter $n$ controls the coarseness of our calibration guarantee.

For any coarseness parameter $n$ and bucket index $i \in [n-1]$, we write $B_n(i) = [\frac{i}{n}, \frac{i+1}{n})$ and $B_n(n) = [\frac{n}{n}, 1]$ so that these buckets partition the unit interval. Conversely, given a $\pi \in [0,1]$, we write $B^{-1}_n(\pi) \in [n]$ for the index of the bucket that $\pi$ belongs to: $B^{-1}_n(\pi) = i$, for $i$ such that $\pi \in B_n(i)$. When clear from the context, we elide the subscript $n$ and write $B(i)$ and $B^{-1}(\pi)$.

For any $S \subseteq [T]$, $i \in [n]$, let $S(i)$ consist of days in $S$ when the mean prediction is in bucket $i$:

$$S(i) = \{ t \in S : \mu_t \in B_n(i) \}.$$

(Simple) calibration asks for the sequence of predictions to be $\alpha$-mean-consistent on all sets $[T](i)$ for $i \in [n]$ – i.e. on all rounds when the prediction was in bucket $i$, for every $i$. Multicalibration asks for the predictions to be calibrated not just on the overall sequence, but also simultaneously on all the subsequences corresponding to each group $G \in \mathcal{G}$. In our notation, it asks for mean consistency on each set $G(i)$, for every group $G \in \mathcal{G}$ and $i \in [n]$.

**Definition 2 (Mean Multicalibration).** A sequence of mean predictions $\{\mu_t\}_{t=1}^T$ is $(\alpha, n)$-mean multicalibrated\(^10\) with respect to $\mathcal{G}$ on a transcript $\pi_T$, if it is $\alpha$-mean-consistent on $G_T(i)$ for all $G \in \mathcal{G}$ and $i \in [n]$:

$$|\mu(G_T(i)) - \mu(G_T(i))| \leq \alpha \frac{T}{|G_T(i)|}.$$

---

\(^9\) Note the scaling of the mean consistency upper bound with both $T$ and $|S|$. If $S = [T]$, then this condition simply asks for the true label mean and the average prediction to be within $\alpha$ of one another, as averaged over the entire transcript. For smaller sets, the allowable error grows with the inverse of $\frac{|S|}{T}$ – i.e. the measure of $S$ within the uniform distribution over the transcript. Even in a distributional setting, estimates inevitably degrade with the size of the set we are conditioning on, and our formulation corresponds exactly to how mean consistency is defined in [6]. Our definitions are also consistent with how the online calibration literature quantifies calibration error with respect to subsequences. [5] handle this issue slightly differently, by asking for uniform bounds, but in the end proving bounds only for sets $S$ that have sufficient mass $\gamma$ in the underlying probability distribution. In the batch setting, our formulation can recover bounds that are strictly stronger than those of [5] after reparameterizing $\alpha \rightarrow \gamma \alpha$.

\(^10\) We define mean multicalibration (and our other notions of multivalidity) to have two parameters: $n$, which controls the coarseness of the guarantee, and $\alpha$, which controls the error of the guarantee. These parameters can be set independently – in the sense that we will be able to achieve $(\alpha, n)$ mean multicalibration for any pair $(\alpha, n)$ – but they should be interpreted together. For example, to avoid the trivial solution in which the learner simply selects uniformly at random at each iteration (thereby guaranteeing that $|G_T(i)| \leq T/n$ for all $G, i$), we should set $\alpha \ll \frac{1}{n}$. 
Interval Predictions

In this case, the prediction domain is the set of ordered pairs of endpoints in the unit interval: \( \mathcal{P}_{\text{interval}} = \{(\ell, u) : \ell \leq u, \; u, \ell \in [0, 1]\} \). Given a pair \((\ell, u) \in \mathcal{P}_{\text{interval}}\), we say that it covers a label \(y \in [0, 1]\) if \(y\) falls between \(\ell\) and \(u\), and write \(\text{Cover}((\ell, u), y) = 1\). We define this coverage indicator as \(\text{Cover}((\ell, u), y) = \mathbb{1}(y \in [\ell, u])\) for \(u < 1\), and \(\mathbb{1}(y \in [\ell, u])\) for \(u = 1\).

In each round \(t\), we will predict an interval \(p_t = (\ell_t, \pi_t)\) with the goal of achieving, for a target coverage probability \(1 - \delta \in [0, 1]\), that \(\mathbb{E}[(\ell_t, \pi_t), y)|x_t| = 1 - \delta\). We again bucket our coverage intervals using a discretization parameter \(n\), using the same notation as for moment predictions.

Towards defining interval prediction consistency, for any \(S \subseteq [T]\) we define

\[
\overline{H}(S) = \frac{1}{|S|} \sum_{t \in S} \text{Cover}((\ell_t, \pi_t), y_t).
\]

\textbf{Definition 3 (Interval Prediction Consistency).} We say that a sequence of interval predictions \(\{(\ell_t, \pi_t)\}_{t=1}^T\) is \(\alpha\)-consistent on set \(S \subseteq [T]\) with respect to failure probability \(\delta \in (0, 1)\), if:

\[
|\overline{H}(S) - (1 - \delta)| \leq \alpha \cdot \frac{T}{|S|}.
\]

For any \(S \subseteq [T]\) and \(i \leq j \in [n]\), we define \(S(i, j)\) to be the subset of rounds in \(S\) in which our predicted interval’s endpoints are in buckets \(i\) and \(j\), respectively. Formally,

\[
S(i, j) = \{t \in S : (\ell_t, \pi_t) \in B_n(i, j)\}.
\]

We can now define multivalidity analogously to how we defined multicalibration.

\textbf{Definition 4 (Interval Prediction Multivalidity).} Given transcript \(\pi_T\), a sequence of interval predictions \(\{(\ell_t, \pi_t)\}_{t=1}^T\) is \((\alpha, n)\)-multivalid with respect to \(\mathcal{G}\), if for every \(1 \leq i \leq j \leq n\) and \(G \in \mathcal{G}\), it is \(\alpha\)-consistent on \(G_T(i, j)\) with respect to coverage probability \(1 - \delta\):

\[
|\overline{H}(G_T(i, j)) - (1 - \delta)| \leq \alpha \cdot \frac{T}{|G_T(i, j)|}.
\]

2.3 The Minimax Theorem

The centerpiece of our analysis will be the minimax theorem for convex-concave two-player zero-sum games.\(^\text{11}\) We briefly recall the setting in which this result holds.

A (convex-concave two-player) zero-sum game is played by a minimization player, with a convex and compact strategy space \(Q_1 \subseteq \mathbb{R}^{d_1}\), and a maximization player, with a convex and compact strategy space \(Q_2 \subseteq \mathbb{R}^{d_2}\) (where \(d_1, d_2 > 0\)). The objective function \(u : Q_1 \times Q_2 \rightarrow \mathbb{R}\) is convex in its first argument and concave in its second argument. In the game, the minimization player chooses some action \(Q_1 \in Q_1\) and the maximization player chooses some action \(Q_2 \in Q_2\), resulting in objective value \(u(Q_1, Q_2)\). The goal of the minimization player is to minimize the objective value, and the goal of the maximization player is to maximize it.

The key property of such zero-sum games, known as the minimax theorem, states that the order of play does not affect the objective value that each player can guarantee.

\textbf{Theorem 5 (Sion’s Minimax Theorem).} For a zero-sum game \((Q_1, Q_2, u)\) as defined above:

\[
\min_{Q_1 \in Q_1} \max_{Q_2 \in Q_2} u(Q_1, Q_2) = \max_{Q_2 \in Q_2} \min_{Q_1 \in Q_1} u(Q_1, Q_2).
\]

\(^{\text{11}}\)In fact, we will use Sion’s extension of the original Von Neumann’s minimax theorem.
Online Multivalid Learning: Means, Moments, and Prediction Intervals

For \((Q_1, Q_2, u)\) a zero-sum game, \(v = \min Q_1 \max Q_2 u(Q_1, Q_2) = \max Q_2 \min Q_1 u(Q_1, Q_2)\) is called its value. A strategy \(Q_1^* \in Q_1\) for the minimization player is a (minimax) equilibrium strategy if it forces the objective to be at most the value of the game, for any strategy \(Q_2 \in Q_2\) of the maximization player: i.e. \(\max_{Q_2} u(Q_1^*, Q_2) = v\). Finally, \(Q_2 \in Q_2\) is the maximization player’s best response to \(Q_1^*\), if it realizes this maximum.

In our analysis, the Learner will be the minimization player and the Adversary the maximization player. Their strategy spaces will be denoted \(Q^L\) and \(Q^A\), respectively. Often, one or both of \(Q^L\) and \(Q^A\) will be a probability simplex over some finite pure action space.\(^{12}\)

### 3 Multicalibration Guarantees for Means and Prediction Intervals

This section states our multivalidity guarantees for mean predictions and interval predictions. These guarantees are of two flavors: ones which hold in expectation over the randomness of the transcript, and ones which hold with high probability over the transcript randomness.

#### Mean Predictions

It will be convenient\(^ {13}\) for us to imagine that the learner’s pure strategy space is a finite, discrete subset of \(P_{\text{mean}} = [0, 1]\). To this end, we define the following discretization for any \(r \in \mathbb{N}\) (here \(n\) is the parameter that represents the coarseness of our bucketing):

\[
{\mathcal{P}^r} = \left\{ \frac{\alpha}{r^n}, \frac{2}{r^n}, \ldots, 1 \right\}.
\]

We use this discretization also in our algorithm in Section 4.2 – but we remark at the outset that the need to discretize is only for technical reasons, and our algorithm will have no dependence – neither in runtime nor in its convergence rate – on the value of \(r\) that we choose, so we can imagine the discretization to be arbitrarily fine.

**Theorem 6 (Multicalibrated Mean Predictions).** When Algorithm 1 is run using \(n\) buckets for calibration, discretization \(r \in \mathbb{N}\), and \(
\eta = \sqrt{\frac{\ln(2|G|n)}{2T}} \in (0, 1/2),
\)
then against any adversary, its sequence of mean predictions is \((\alpha, n)\)-multicalibrated with respect to \(G\), where:\(^ {14}\)

\[
\begin{align*}
\mathbb{E}[\alpha] &\leq \frac{1}{\eta^2} + 2\sqrt{\frac{2}{\eta} \ln(2|G|n)}.
\end{align*}
\]

In particular, we can guarantee\(^ {15}\) \(\mathbb{E}[\alpha] \leq (2 + \epsilon) \sqrt{\frac{2}{\eta} \ln(2|G|n)}\).

\[
\begin{align*}
\alpha &\leq \frac{1}{\eta^2} + 4\sqrt{\frac{2}{\eta} \ln \left(\frac{2|G|n}{\lambda}\right)}.
\end{align*}
\]

In particular, we can guarantee\(^ {16}\) \(\alpha \leq (4 + \epsilon) \sqrt{\frac{2}{\eta} \ln \left(\frac{2|G|n}{\lambda}\right)}\).

\(^{12}\)Zero-sum games are often defined by endowing each player with a finite set of pure strategies \(X_1, X_2\). The convex compact strategy sets \(Q_1\) and \(Q_2\) are then formed by allowing players to randomize over their pure strategies and taking \(Q_1 = \Delta X_1, \ Q_2 = \Delta X_2\) to be the probability simplices over the pure strategies of each player. An objective function \(u : X_1 \times X_2 \to \mathbb{R}\) can be linearly extended to \(Q_1\) and \(Q_2\) in the natural way (by taking expectations over the randomized strategies of each player) – i.e. for any \(Q_1 \in Q_1\) and \(Q_2 \in Q_2\), we write \(u(Q_1, Q_2) = \mathbb{E}_{x_1 \sim Q_1, x_2 \sim Q_2}[u(x_1, x_2)]\).

\(^{13}\)In order to satisfy the convexity and compactness requirements of the minimax theorem.

\(^{14}\)In both bounds below, the dependence on \(\log(|G|)\) can be replaced with a dependence on \(\log(d)\) under the assumption that \(|G(x)| \leq d\) for all \(x\) – i.e. that each observed data point is contained in only boundedly many groups. This gives us non-trivial guarantees even when \(G\) is infinitely large.

\(^{15}\)For any \(\epsilon > 0\), by choosing \(r = \sqrt{T/(\epsilon n \sqrt{2 \ln(2|G|n)})}\).

\(^{16}\)For any \(\epsilon > 0\), by choosing \(r = \sqrt{T/(\epsilon n \sqrt{2 \ln(2|G|n/\lambda)})}\).
Prediction Intervals

We now describe our existential multivalidity guarantees, with respect to any collection of groups \( \mathcal{G} \), for the online adversarial setting of prediction intervals with a coverage target \( 1 - \delta \). When \( \mathcal{G} = \{ \mathcal{X} \} \), this setting reduces to giving simple marginal prediction intervals—a similar problem as solved by conformal prediction\(^\text{17} \), but without requiring distributional assumptions. For richer classes \( \mathcal{G} \), our guarantees correspondingly become stronger.

Before proceeding, we pause to observe that even in the easier distributional setting where data are drawn from a fixed distribution: \( (x, y) \sim D \) there may not be any interval \( (\ell, u) \in \mathcal{P}_{\text{interval}} \) that satisfies the desired target coverage value, i.e., that guarantees that \( \Pr_{(x,y) \sim D} [\text{Cover}(\ell, u, y)] - (1 - \delta) \) is small. Consider for example a label distribution that places all its mass on a single value \( y = i \in [0, 1] \). Then any interval \( (\ell, u) \) covers the label with probability 1 or probability 0, which for \( \delta \not\in \{0, 1\} \) is bounded away from our target coverage probability. Of course, if achieving the target coverage is impossible in the easier distributional setting, then it is also impossible in the more challenging online adversarial setting. With this in mind, we define a class of smooth distributions for which achieving approximate target coverage is always possible for some interval \( (\ell, u) \) defined over an appropriately finely discretized range:

\[
\mathcal{P}^\rho_{\text{interval}} = \{(i, j) \in \mathcal{P}_{\text{interval}} : i, j \in \mathcal{P}^\rho\} \quad \text{(where } \mathcal{P}^\rho = \{0, 1/(\rho n), \ldots, 1\}\).
\]

Formally, a label distribution \( Q \in \Delta \mathcal{Y} \) is called \((\rho, \rho n)\)-smooth if

\[
\Pr_{y \sim Q} [y \in [a, b]] \leq \rho
\]

for any \( 0 \leq a \leq b \leq 1 \) with \( |a - b| \leq \frac{1}{\rho n} \). A joint distribution \( D \in \Delta(\mathcal{X} \times \mathcal{Y}) \) is called \((\rho, \rho n)\)-smooth if for every \( x \in \mathcal{X} \), the label distribution conditional on \( x \), \( Q|_{x} \), is \((\rho, \rho n)\)-smooth.

Intuitively, a smooth distribution is not overly concentrated (has mass \( \leq \rho \)) on any sub-interval of width \( 1/(\rho n) \), allowing \( \rho \)-approximate coverage by \( 1/(\rho n) \)-discretized intervals.

> Observation 3.1. For any \( \delta \in [0, 1] \) and any fixed \((\rho, \rho n)\)-smooth label distribution \( Q \), there always exists some interval \( (\bar{\ell}, \bar{u}) \in \mathcal{P}^\rho_{\text{interval}} \) such that \( \Pr_{y \sim Q} [\text{Cover}((\bar{\ell}, \bar{u}, y)] - (1 - \delta) \leq \rho \).

We show that we can similarly achieve (approximately) our target coverage goals in the online adversarial setting when the adversary is constrained to playing smooth distributions.

> Remark 7. The assumption of \((\rho, \rho n)\)-smoothness becomes more mild for any \( \rho \) as \( r \to \infty \). Just as for mean multicalibration, in which our error bounds inevitably depend on the chosen discretization level \( r \), here our error bounds will depend on the smoothness level \( \rho \) of the adversary’s distributions at the chosen value of \( r \).

However, we note that smoothness is an extremely mild condition, as we can enforce it ourselves if we so choose, rather than assuming the adversary is constrained. This is easily achieved\(^\text{18} \) e.g. by perturbing observed levels with uniform noise from \([-\epsilon, \epsilon]\); when we do this, the intervals we obtain continue to have valid coverage if we widen both endpoints by \( \epsilon \).

\(^{17}\)In fact, even with \( \mathcal{G} = \{ \mathcal{X} \} \) the guarantees are stronger than the marginal guarantees promised by conformal prediction techniques, because they remain valid even conditioning on the prediction. This is important and rules out trivial solutions, like predicting the full interval with probability 1 - \( \delta \) and an empty interval with probability \( \delta \).

\(^{18}\)See the argument in the full version of this paper.
Theorem 8 (Multivalid Prediction Intervals). When Algorithm 3 is run using \( n \) buckets, discretization parameter \( r \) and \( \eta = \sqrt{\frac{\ln(2|\mathcal{G}|n^2)}{2T}} \in (0, 1/2) \), then against any adversary constrained to playing \((\rho, rn)\)-smooth distributions, its sequence of interval predictions is \( \alpha \)-multivalid with respect to \( \mathcal{G} \), where:

- **(In-Expectation Guarantee)** In expectation over the randomness of the transcript \( \pi_T \),
  \[ \mathbb{E}[\alpha] \leq \rho + 2 \sqrt{\frac{2 \ln(2|\mathcal{G}|n^2)}{T}}. \]

- **(High-Probability Guarantee)** With probability \( 1 - \lambda \) over the transcript randomness,
  \[ \alpha \leq \rho + 4 \sqrt{\frac{2}{T} \ln \left( \frac{2|\mathcal{G}|n^2}{\lambda} \right)} . \]

## Online Mean Multicalibration

In this section, we show how to obtain mean multicalibrated estimators in an online adversarial setting. Our derivation also serves as a warm up example of our general technique, which we also instantiate (in somewhat more involved settings) for multivalid prediction intervals and mean-moment multicalibration. At a high level, the derivation of our algorithm and its proof of correctness proceeds as follows:

1. For each \( G \in \mathcal{G} \), \( i \in [n] \), \( s \in [T] \), and \( \pi_{s} \), we define an empirical quantity \( V_{s}^{G,i} \) (Definition 9) which represents the calibration error incurred by our algorithm on group \( G \) over those of the rounds \( 1 \ldots s \) when the \( i \)th bucket was predicted. If \( |V_{s+1}^{G,i}| \) is small for all \( G, i \), then our algorithm is approximately multicalibrated with respect to \( \mathcal{G} \) after \( T \) rounds.

   The premise of our algorithm is to greedily make the decision at each round \( s \) so as to minimize the maximum possible increase \( \max_{G,i} |V_{s+1}^{G,i}| - \max_{G,i} |V_{s}^{G,i}| \) of these quantities, in the worst case over the choices of the adversary. If we could bound this quantity at every round, then by telescoping, we would have a bound on \( \max_{G,i} |V_{T}^{G,i}| \) at the end of the interaction, and therefore a guarantee of mean multicalibration.

2. The increase in the maximum value of \( |V_{s+1}^{G,i}| \) is inconvenient to work with, and so we instead define a smooth potential function \( L_{s} \) (Definition 10) corresponding to a soft-max function which upper bounds \( \max_{G,i} |V_{s}^{G,i}| \). Our design goal instead becomes to upper bound the increase in our potential function from round to round: \( \Delta_{s+1} = L_{s+1} - L_{s} \).

   We view this as defining a zero-sum game, in which the learner’s goal is to minimize this increase, and the adversary’s goal is to maximize it.

3. We show that for each fixed distribution that the adversary could employ at each round \( s + 1 \), there is a prediction the learner could employ (if only she knew the adversary’s distribution) that would guarantee that the increase in potential \( \Delta_{s+1} \) is small. Intuitively, this is because if we knew the true joint distribution over feature label pairs, then we could predict the true conditional expectations, \( \pi_{s+1} = \mathbb{E}[y_{s+1}|x_{s+1}] \), which would be perfectly calibrated on all groups. Of course, the learner does not have the luxury of knowing the adversary’s distribution before choosing her own. But this thought experiment establishes the value of the game, and so we can conclude via the minimax theorem that there must be some fixed distribution over prediction rules that the learner can play that will guarantee \( \Delta_{s+1} \) being small against all actions of the adversary.

4. Step 3 suffices to argue for the existence of an algorithm obtaining multicalibration guarantees (Algorithm 1). However, to actually derive an implementable algorithm we need to find a way to compute the equilibrium strategy at each round, whose existence
was argued in Step 3. A priori, this seems daunting because the learner’s strategy space consists of all randomized mappings between \( X \) and \( Y \), and the adversary’s strategy space consists of all joint distributions on \( X \times Y \). However, we derive a simple algorithm in Section 4.2 that implements the optimal equilibrium strategy needed to realize Step 3. Informally, we are able to do so by representing the mapping between \( X \) and \( Y \) only implicitly, and delaying all computation until \( x_t \) has been chosen. We then show that the equilibrium strategy for the learner has a simple structure and randomizes over only at most 2 predictions. Our final algorithm (Algorithm 2) simply computes the relevant portion of the equilibrium strategy at each round and then samples from it.

4.1 An Existential Derivation of the Algorithm and Multicalibration Bounds

We begin by defining notation \( V_{G,i}^{G,i} \) for the (unnormalized) portion of the mean calibration error corresponding to each group \( G \in \mathcal{G} \) and bucket \( i \in [n] \):

\[
V_{G,i}^{G,i}(\pi_s) = \left| G_s(i) \right| (\mu_s(G_s(i)) - \overline{y}_{G_s(i)}) = \sum_{t=1}^{s} \mathbb{I}[y_t \in B(i), x_t \in G_s] (y_t - \overline{y}_t)
\]

When the transcript is clear from context we will sometimes simply write \( V_{G,i}^{G,i} \).

Observe that our definition of mean multicalibration (Definition 2) corresponds to asking that \( \left| V_{G,i}^{G,i} \right| \) be small for all \( i,G \).

\[
\text{Definition 9.} \quad \text{Given a transcript } \pi_s = ((x_t, \overline{y}_t, y_t))_{t=1}^{s}, \text{ we define the mean calibration error for a group } G \in \mathcal{G} \text{ and bucket } i \in [n] \text{ at time } s \text{ to be:}
\]

\[
V_{G,i}^{G,i}(\pi_s) = \left| G_s(i) \right| (\mu_s(G_s(i)) - \overline{y}_{G_s(i)}) = \sum_{t=1}^{s} \mathbb{I}[y_t \in B(i), x_t \in G_s] (y_t - \overline{y}_t)
\]

\[
\text{Observe that this “soft-max style” function allows us to tightly upper bound our calibration loss:}
\]

\[
\text{Definition 10 (Surrogate loss function).} \quad \text{Fixing a transcript } \pi_s \in \Pi^* \text{ and a parameter } \eta \in [0, \frac{1}{2}], \text{ define a surrogate calibration loss function at day } s \text{ as:}
\]

\[
L_s(\pi_s) = \sum_{G \in \mathcal{G}, i \in [n]} \left( \exp(\eta V_{G,i}^{G,i}) + \exp(-\eta V_{G,i}^{G,i}) \right).
\]

\[
\text{When the transcript } \pi_s \text{ is clear from context, we will sometimes simply write } L_s.
\]

We will leave \( \eta \) unspecified for now, and choose it later to optimize our bounds. Observe that this “soft-max style” function allows us to tightly upper bound our calibration loss:

\[
\text{Observation 4.2.} \quad \text{For any transcript } \pi_T, \text{ and any } \eta \in [0, \frac{1}{2}], \text{ we have that:}
\]

\[
\max_{G \in \mathcal{G}, i \in [n]} \left| V_{G,i}^{G,i} \right| \leq \frac{1}{\eta} \ln(L_T) \leq \max_{G \in \mathcal{G}, i \in [n]} \left| V_{G,i}^{G,i} \right| + \frac{\ln(2|\mathcal{G}|n)}{\eta}.
\]

Part of our analysis will depend on viewing the transcript as a random variable: in this case, in keeping with our convention for random variables, we refer to it as \( \tilde{\pi} \). The associated random variables tracking calibration and surrogate loss are denoted \( \tilde{V} \) and \( \tilde{L} \) respectively.
Our goal is to find a strategy for the learner that keeps our surrogate loss $L_s$ small. Towards this end, we define $\Delta_{s+1}(\pi_s, x_{s+1}, \bar{\pi}_{s+1})$ to be the expected increase in the surrogate loss in the event that the adversary plays feature vector $x_{s+1}$ and the learner plays prediction $\bar{\pi}_{s+1}$. The expectation is over the only remaining source of randomness after the conditioning – the distribution over labels $y_{s+1}$ (which is fully determined by $\pi_s$ and $x_{s+1}$).

**Definition 11 (Conditional Change in Surrogate Loss).**

$$\Delta_{s+1}(\pi_s, x_{s+1}, \bar{\pi}_{s+1}) = \mathbb{E}_{\bar{y}_{s+1}} \left[ \bar{L}_{s+1} - L_s | x_{s+1}, \bar{\pi}_{s+1}, \pi_s \right].$$

We begin with a simple bound on $\Delta_{s+1}(\pi_s, x_{s+1}, \bar{\pi}_{s+1})$. For notational convenience define, for all $i \in [n]$, the quantity $C_i^s(x_{s+1}) \equiv \sum_{G(x_{s+1})} \exp(\eta V_{s}^{G,i}) - \exp(-\eta V_{s}^{G,i}).$

**Lemma 12.** For any $\pi_s \in \Pi^*$, $x_{s+1} \in \mathcal{X}$, $\bar{\pi}_{s+1} \in \mathcal{P}_{\text{mean}}$, and $i \in [n]$ such that $\bar{\pi}_{s+1} \in B(i)$:

$$\Delta_{s+1}(\pi_s, x_{s+1}, \bar{\pi}_{s+1}) \leq \eta \left( \mathbb{E}_{\bar{y}_{s+1}} [\bar{y}_{s+1} - \bar{\pi}_{s+1}] \right) C_i^s(x_{s+1}) + 2\eta^2 L_s.$$

**Proof.** Fix any transcript $\pi_s \in \Pi^*$ (which defines $L_s$), feature vector $x_{s+1} \in \mathcal{X}$, and $\bar{\pi}_{s+1}$ such that $\bar{\pi}_{s+1} \in B(i)$ for some $i \in [n]$. By direct calculation, we obtain:


given $\Delta_{s+1}(\pi_s, x_{s+1}, \bar{\pi}_{s+1})$

$$\Delta_{s+1}(\pi_s, x_{s+1}, \bar{\pi}_{s+1}) = \mathbb{E}_{\bar{y}_{s+1}} \left[ \sum_{G(x_{s+1})} \exp(\eta V_{s}^{G,i})(\exp(\eta(\bar{y}_{s+1} - \bar{\pi}_{s+1})) - 1) + \exp(-\eta V_s^{G,i})(\exp(-\eta(\bar{y}_{s+1} - \bar{\pi}_{s+1})) - 1) \right],$$

$$\leq \mathbb{E}_{\bar{y}_{s+1}} \left[ \sum_{G(x_{s+1})} \exp(\eta V_{s}^{G,i})(\eta(\bar{y}_{s+1} - \bar{\pi}_{s+1}) + 2\eta^2) + \exp(-\eta V_s^{G,i})(\eta(\bar{y}_{s+1} - \bar{\pi}_{s+1}) + 2\eta^2) \right],$$

$$= \eta \left( \mathbb{E}_{\bar{y}_{s+1}} [\bar{y}_{s+1} - \bar{\pi}_{s+1}] \right) \sum_{G(x_{s+1})} \left[ \exp(\eta V_{s}^{G,i}) - \exp(-\eta V_s^{G,i}) \right] + 2\eta^2 \sum_{G(x_{s+1})} \left[ \exp(\eta V_{s}^{G,i}) + \exp(-\eta V_s^{G,i}) \right],$$

$$\leq \eta \left( \mathbb{E}_{\bar{y}_{s+1}} [\bar{y}_{s+1} - \bar{\pi}_{s+1}] \right) \left( \sum_{G(x_{s+1})} \exp(\eta V_{s}^{G,i}) - \exp(-\eta V_s^{G,i}) \right) + 2\eta^2 L_s,$n

$$= \eta \left( \mathbb{E}_{\bar{y}_{s+1}} [\bar{y}_{s+1} - \bar{\pi}_{s+1}] \right) C_i^s(x_{s+1}) + 2\eta^2 L_s.$$

Here, the first inequality follows from the fact that for $0 < |x| < \frac{1}{2}$, $\exp(x) \leq 1 + x + 2x^2$. ☐

Using this bound, we will shortly define a zero-sum game between the learner and the adversary and use the minimax theorem to conclude that the learner always has a strategy that guarantees that the per-round increase in surrogate loss can be bounded.

Before proceeding, recall (from Section 3) our notation for the Learner’s strategy space: \(\mathcal{P}_{\text{str}} = \{0, 1, \ldots, \pi_{n}, \ldots, 1\}\). Also, for notational convenience, for each $\bar{\pi} \in \mathcal{P}_{\text{str}}$, define $C_i^\pi \equiv C_i$ where $i \in [n]$ s.t. $\bar{\pi} \in B_n(i)$.

**Lemma 13.** For any transcript $\pi_s \in \Pi^*$, any $x_{s+1} \in \mathcal{X}$, and any $r \in \mathbb{N}$ there exists a distribution over predictions for the learner $Q_{\pi_s}^L \in \Delta\mathcal{P}_{\text{str}}$, such that regardless of the adversary’s choice of distribution of $y_{s+1}$ over $\Delta\mathcal{Y}$, we have that:

$$\mathbb{E}_{\bar{\pi} \sim Q_{\pi_s}^L} [\Delta_{s+1}(\pi_s, x_{s+1}, \bar{\pi})] \leq L_s \left( \frac{\eta}{rn} + 2\eta^2 \right).$$
Proof. We define a zero-sum game played between the learner (the minimization player) and the adversary (the maximization player). The learner’s pure strategy space is the set of discrete predictions $X_1 = \mathcal{P}^n$. The adversary’s pure strategy space is (a priori) the set of all distributions over labels in $[0, 1]$. However, we will observe in a moment that the objective function of our game depends only on the expected value of the label, and so without loss of generality, we will be able to take the adversary’s full strategy space to be the set of all pure strategies, i.e., $\mathcal{Q}^1 = [0, 1]$ (which is closed and convex), because it already spans the set of realizable expectations. As usual, we take the learner’s full strategy space to be the set of distributions over pure strategies: $\mathcal{Q}^L = \Delta \mathcal{P}^n$.

Fix the transcript $\pi_s$ and the feature vector $x_{s+1}$. We define the objective of this game as the upper bound on $\Delta_{s+1}(\pi_s, x_{s+1}, \overline{\pi})$ proved in Lemma 12. For each $\pi \in \mathcal{P}^n$, $y \in [0, 1]$, let:

$$u(\pi, y) \equiv \eta (y - \overline{\pi}) C^T(x_{s+1}) + 2\eta^2 L_s.$$

Note that for any distribution over labels $y$ of the adversary, the expected objective value depends on his strategy only through $\mathbb{E}[\tilde{y}]$ because the above objective function is linear in $y$; that is, $\mathbb{E}_Q[u(\tilde{y}, \tilde{y})] = u(\overline{\pi}, \mathbb{E}[\tilde{y}])$. Thus we are justified in our reduced-form representation of the adversary’s full strategy as choosing $\mathbb{E}[\tilde{y}]$ in the interval $[0, 1]$.

Now, observe that for any strategy of the adversary (which fixes $\mathbb{E}[\tilde{y}]$), the learner can respond by playing $\overline{\pi} = \mathop{\text{argmin}}_{\pi' \in \mathcal{P}^n} \mathbb{E}[\tilde{y}] - \pi'$, and that because of our discretization, $\min |\mathbb{E}[\tilde{y}] - \overline{\pi}| \leq \frac{1}{n}$. Therefore, the value of the game is at most:

$$\max_{y \in [0, 1]} \min_{\pi \in \mathcal{P}^n} u(\pi, y) \leq \frac{\eta}{rn} \left( C^T(x_{s+1}) + 2\eta^2 L_s \right) \leq L_s \left( \frac{\eta}{rn} + 2\eta^2 \right).$$

Here the latter inequality follows since $C^T(x_{s+1}) \leq L_s$ for all $\pi \in \mathcal{P}^n$, by observation. We can now apply the minimax theorem (Theorem 5) to conclude that there exists a fixed distribution $Q^L_{s+1} \in \mathcal{Q}^L$ for the learner that guarantees that simultaneously for every label $y \in [0, 1]$ that might be chosen by the adversary, we have the desired bound

$$\mathbb{E}_{\pi \sim Q^L_{s+1}} [u(\pi, y)] \leq L_s \left( \frac{\eta}{rn} + 2\eta^2 \right).$$

**Corollary 14.** For every $r \in \mathbb{N}$, $s \in [T]$, $\pi_s \in \Pi^*$, and $x_{s+1} \in \mathcal{X}$ (which fixes $L_s$ and $Q^L_{s+1}$), and any distribution over $\mathcal{Y}$:

$$\mathbb{E}_{\pi_{s+1} \sim Q^L_{s+1}} [\Delta_{s+1} | \pi_s] = L_s + \mathbb{E}_{\pi_{s+1} \sim Q^L_{s+1}} [\Delta_{s+1}(\pi_s, x_{s+1}, \overline{\pi}_{s+1})] \leq L_s \left( 1 + \frac{\eta}{rn} + 2\eta^2 \right).$$

Lemma 13 defines the following generic Algorithm 1, which the learner can use to make predictions. First, we show that if we can compute the distributions $Q^L_t$, Algorithm 1 will produce multicalibrated predictions. Next, in Section 4.2 we show a simple and efficient method for sampling from $Q^L_t$.

**Algorithm 1** A Generic Multicalibrator.

```
for $t = 1, \ldots, T$
    Observe $x_t$. Given $\pi_{t-1}$ and $x_t$, let $Q^L_t \in \mathcal{Q}^L_t$ be the distribution over predictions whose existence is established in Lemma 13.
    Sample $\overline{\pi} \sim Q^L_t$ and predict $\overline{\pi}_t = \overline{\pi}$
```

We now prove two convergence bounds for Algorithm 1. The first bounds its multicalibration error in expectation, and the other provides a high probability bound. To show these bounds, we first state a helper theorem (which we also use to prove our bounds in the prediction intervals setting and in the mean-moment setting).
Theorem 15. Consider a random process $\tilde{X}_t \geq 0$ adapted to the filtration $\mathcal{F}_t = \sigma(\pi_t)$, where $\tilde{X}_0$ is constant a.s. Suppose that for every period $t$ and transcript $\pi_{t-1}$, it holds that

$$E[\tilde{X}_t|\pi_{t-1}] \leq X_{t-1}(1 + \eta c + 2\eta^2)$$

for some $\eta \in [0, \frac{1}{2}], c \in [0, 1]$. Then:

$$E[\tilde{X}_T] \leq X_0 \exp (T\eta c + 2T\eta^2). \quad (2)$$

Further, define a process $\tilde{Z}_t$ adapted to the same filtration by $\tilde{Z}_t = Z_{t-1} + \ln \tilde{X}_t - E[\ln(\tilde{X}_t)|\pi_{t-1}]$. Suppose that $|Z_t - Z_{t-1}| \leq 2\eta$, where $Z_0 = 0$ a.s. Then, with probability $1 - \lambda$,

$$\ln(X_T(\pi_T)) \leq \ln(X_0) + T(\eta c + 2\eta^2) + \eta \sqrt{8T\ln \left(\frac{1}{\lambda}\right)}. \quad (3)$$

The first part of Theorem 15 follows by a telescoping argument and the tower rule of expectation. The second part is derived via the Azuma inequality for martingales.

Proof of the in-expectation bound of Theorem 6. We are now ready to bound our multi-calibration error. By Corollary 14 and the first part of Theorem 15 (with $L_0 = 2|\mathcal{G}| n$ and $c = \frac{1}{rn}$ plugged in), we have that Algorithm 1 results in the surrogate loss satisfying

$$E[\tilde{L}_T] \leq 2|\mathcal{G}| n \exp \left(\frac{Tn}{rn} + 2T\eta^2\right). \quad (4)$$

Next, we can convert this into a bound on Algorithm 1’s expected calibration error. Indeed, by Observation 4.1, it suffices to show

$$\frac{1}{T} \mathbb{E}_{\pi_T} \left[ \max_{G \in |\mathcal{G}|, c \in |n|} |\bar{V}_{T}^{G,i}| \right] \leq \frac{1}{rn} + 2\sqrt{\frac{2\ln(2|\mathcal{G}| n)}{T}}. \quad$$

We begin by computing a bound on the (exponential of) the expectation of this quantity:

$$\exp \left(\eta \mathbb{E}_{\pi_T} \left[ \max_{G,i} |\bar{V}_{T}^{G,i}| \right] \right) \leq \mathbb{E}_{\pi_T} \left[ \exp \left(\eta \max_{G,i} |\bar{V}_{T}^{G,i}| \right) \right],$$

$$= \mathbb{E}_{\pi_T} \left[ \max_{G,i} \left( \eta |\bar{V}_{T}^{G,i}| \right) \right],$$

$$\leq \mathbb{E}_{\pi_T} \left[ \sum_{G,i} \left( \exp \left(\eta |\bar{V}_{T}^{G,i}| \right) + \exp \left(-\eta |\bar{V}_{T}^{G,i}| \right) \right) \right],$$

$$= \mathbb{E}_{\pi_T} \left[ \bar{L}_T \right],$$

$$\leq 2|\mathcal{G}| n \exp \left(\frac{Tn}{rn} + 2T\eta^2\right).$$

The first step uses Jensen’s inequality and the last one is by Bound 4. Taking the logarithm of both sides, dividing by $\eta T$, and choosing $\eta = \sqrt{\frac{\ln(2|\mathcal{G}| n)}{2T}}$, gives the desired result:

$$\frac{1}{T} \mathbb{E}_{\pi_T} \left[ \max_{G,i} |\bar{V}_{T}^{G,i}| \right] \leq \frac{\ln(2|\mathcal{G}| n)}{\eta T} + \frac{1}{rn} + 2\eta = \frac{1}{rn} + 2\sqrt{\frac{2\ln(2|\mathcal{G}| n)}{T}}. \quad \blacktriangle$$

Proof of the high-probability bound of Theorem 6. Given $\bar{L}$, suppose $\hat{Z}$ is its associated martingale process as defined in the second part of Theorem 15. Then, it is easy to show that the increments of $\hat{Z}$ are uniformly bounded over all rounds $t$ — that is, at any round $t$
and for any realized transcript \( \pi_t \), we have \( |Z_t - Z_{t-1}| \leq 2\eta \). As a result, the second part of Theorem 15 applies; plugging in \( L_0 = 2|G|n \) and \( c = \frac{1}{r_n} \), we have:

\[
\ln(L_T(\pi_T)) \leq \ln(2|G|n) + T \left( \frac{\eta}{r_n} + 2\eta^2 \right) + \eta \sqrt{8T \ln \left( \frac{1}{\lambda} \right)}.
\]

Now, note that

\[
\exp \left( \eta \max_{G,i} \left| V_{T,i}^G \right| \right) = \max_{G,i} \exp \left( \eta V_{T,i}^G \right),
\]

\[
\leq \max_{G,i} \left( \exp \left( \eta V_{T,i}^G \right) + \exp \left( -\eta V_{T,i}^G \right) \right),
\]

\[
\leq \sum_{G,i} \left( \exp \left( \eta V_{T,i}^G \right) + \exp \left( -\eta V_{T,i}^G \right) \right),
\]

\[
= L_T(\pi_T).
\]

Taking log on both sides and dividing both sides by \( \eta T \), we get

\[
\frac{1}{T} \max_{G,i} |V_{T,i}^G| \leq \frac{1}{\eta T} \ln(L_T(\pi_T)) \leq \frac{\ln(2|G|n)}{\eta T} + \frac{1}{r_n} + 2\eta + \sqrt{\frac{8 \ln(\frac{1}{\lambda})}{T}}.
\]

Choosing \( \eta = \sqrt{\frac{\ln(2|G|n)}{2T}} \), we thus obtain the desired inequality

\[
\frac{1}{T} \max_{G,i} |V_{T,i}^G| \leq \frac{1}{r_n} + 2 \sqrt{\frac{2 \ln(2|G|n)}{T}} + \sqrt{\frac{8 \ln(\frac{1}{\lambda})}{T}} \leq \frac{1}{r_n} + 4 \sqrt{\frac{2 \ln(\frac{1}{\lambda})}{T}}.
\]

4.2 Deriving an Efficient Algorithm via Equilibrium Computation

In Section 4.1, we derived Algorithm 1 and proved that it results in mean multicalibrated predictions. However, Algorithm 1 was not defined explicitly: it relies on the distributions \( Q_t^k \), whose existence we showed in Lemma 13 but which we did not explicitly construct. In this section, we derive a scheme for sampling from these distributions \( Q_t^k \), which leads to Algorithm 2 — an explicit, efficient implementation of Algorithm 1.

\textbf{Algorithm 2} Von Neumann’s Mean Multicalibrator(\( \eta, n, r \)).

\begin{algorithm}
\begin{algorithmic}
  \State \textbf{for} \( t = 1, \ldots, T \) \textbf{do}
  \State Observe \( x_t \). Compute \( C_t^{i-1}(x_t) \equiv \sum_{\varphi(x_t)} \exp(\eta V_{t-1}^G) - \exp(-\eta V_{t-1}^G) \) for \( i \in [n] \).
  \State if \( C_t^{i-1}(x_t) > 0 \) for all \( i \in [n] \) \textbf{then}
  \State Predict \( \overline{\pi}_t = 1 \).
  \State else if \( C_t^{i-1}(x_t) < 0 \) for all \( i \in [n] \) \textbf{then}
  \State Predict \( \overline{\pi}_t = 0 \).
  \State else
  \State Find \( i^* \in [n - 1] \) such that \( C_t^{i^*-1}(x_t) \cdot C_t^{i^*-1}(x_t) \leq 0 \)
  \State Define \( 0 \leq q_t \leq 1 \) such that \( q_t C_t^{i^*-1}(x_t) + (1 - q_t) C_t^{i^*-1}(x_t) = 0 \). That is, let: \( q_t = \frac{|C_t^{i^*-1}(x_t)|}{|C_t^{i^*-1}(x_t)| + |C_t^{i^*-1}(x_t)|} \).
  \State Predict \( \overline{\pi}_t = \frac{i^*}{n} - \frac{1}{n} \) with probability \( q_t \) and \( \overline{\pi}_t = \frac{i^*}{n} \) with probability \( 1 - q_t \).
\end{algorithmic}
\end{algorithm}
Theorem 16. Algorithm 2 implements Algorithm 1, and obtains the multicalibration guarantees of Theorem 6.

Proof. At every round $s+1$, Algorithm 1 samples from a distribution $Q^L_{s+1}$ that is a minimax equilibrium strategy of a game between the learner and the adversary, with objective function

$$u(\pi, y) = \eta (y - \pi) C^\pi_s(x_{s+1}) + 2\eta^2 L_s.$$ 

The equilibrium structure of the game is preserved under positive affine transformations, so instead consider

$$u(\pi, y) = (y - \pi) C^\pi_s(x_{s+1}).$$

We wish to find a distribution $Q^L_{s+1} \in Q^L$ that guarantees – against any strategy of the adversary – an objective value that is at most the bound on the value of the game we proved in Lemma 13. For the transformed game, this bound is:

$$\max_{y \in [0,1]} \mathbb{E}_{\pi \sim Q_s} [u(\pi, y)] \leq \frac{1}{rn} L_s.$$ 

We can start by characterizing the best response of the adversary.

Claim 17. Denote $(x)^+ = \max(x, 0)$. For any $Q^L \in Q^L$:

$$\max_{y \in [0,1]} \mathbb{E}_{\pi \sim Q^L} [u(\pi, y)] = \left( \mathbb{E}_{\pi \sim Q^L} [C^\pi_s(x_{s+1})] \right)^+ - \mathbb{E}_{\pi \sim Q^L} [\pi C^\pi_s(x_{s+1})].$$

Proof. Note that $u(\mu, y) = (y - \mu) C^\pi_s(x_{s+1}) = y C^\pi_s(x_{s+1}) - \pi C^\pi_s(x_{s+1})$. Observe that only the first term depends on $y$. Therefore, if the learner plays according to $Q^L$, then the adversary will choose $y$ so as to maximize the linear expression $y \mathbb{E}_{\pi \sim Q^L} [C^\pi_s(x_{s+1})]$. This is maximized at $y = 1$ when $\mathbb{E}_{\pi \sim Q^L} [C^\pi_s(x_{s+1})] > 0$, and at $y = 0$ otherwise.

Finally, we can reduce the analysis to three disjoint cases:

1. $C^\pi_s(x_{s+1}) > 0$ for all $i \in [n]$: Then for any distribution $Q^L$, by Observation 17 we have:

$$\max_{y \in [0,1]} \mathbb{E}_{\pi \sim Q^L} [u(\pi, y)] = \mathbb{E}_{\pi \sim Q^L} [C^\pi_s(x_{s+1})] - \mathbb{E}_{\pi \sim Q^L} [\pi C^\pi_s(x_{s+1})].$$

In this case, letting $Q^L$ be a point mass on $\pi = 1$ achieves a value of $0 < \frac{1}{rn} L_s$.

2. $C^\pi_s(x_{s+1}) < 0$ for all $i \in [n]$: Then for any distribution $Q^L$, by Observation 17 we have:

$$\max_{y \in [0,1]} \mathbb{E}_{\pi \sim Q^L} [u(\pi, y)] = - \mathbb{E}_{\pi \sim Q^L} [\pi C^\pi_s(x_{s+1})]$$

In this case, letting $Q^L$ be a point mass on $\pi = 0$ achieves a value of $0 < \frac{1}{rn} L_s$.

3. In the remaining case, there must exist some index $i^* \in [n-1]$ such that either $C^\pi_s(x_{s+1})$ and $C^{\pi^*+1}(x_{s+1})$ have opposite signs, or such that at least one of them takes value exactly zero. Randomizing as in the algorithm results in:

$$\max_{y \in [0,1]} \mathbb{E}_{\pi \sim Q^L_{s+1}} [u(\pi, y)] = \left( \mathbb{E}_{\pi \sim Q^L_{s+1}} [C^\pi_s(x_{s+1})] \right)^+ - \mathbb{E}_{\pi \sim Q^L_{s+1}} [\pi C^\pi_s(x_{s+1})]$$

$$= \left[ q_{s+1} C^\pi_s(x_{s+1}) + (1 - q_{s+1}) C^{\pi^*+1}(x_{s+1}) \right]^+$$

$$- \left[ q_{s+1} \left( \frac{\eta}{rn} - \frac{1}{rn} \right) C^\pi_s(x_{s+1}) + (1 - q_{s+1}) \frac{\eta}{rn} C^{\pi^*+1}(x_{s+1}) \right]$$

$$= \frac{1}{rn} C^\pi_s(x_{s+1}) \leq \frac{1}{rn} L_s.$$ 

Algorithm 2 plays this distribution $Q^L_{s+1}$ at every round. This completes the proof.
Running Time

Our algorithm is elementary, and given values for \( C_{t-1}(x_t) \), it runs in time per iteration which is linear in the number of buckets \( n \). For large collections of groups \( \mathcal{G} \), the bulk of the computational cost is due to the first step of Algorithm 2, in which we compute the quantities \( C_{t-1}(x_t) = \sum_{G(x_t)} \exp(\eta V_{G,i}^{G}) - \exp(-\eta V_{G,i}^{G}) \). These quantities sum over every group \( G \in \mathcal{G} \) such that \( x_t \in G \). We can compute this sum in time linear in \( |G| \) by enumerating over all such groups. However, for any class \( G \) such that we can efficiently enumerate the set of groups containing \( x_t \) (i.e. \( \mathcal{G}(x_t) \)), our per-round runtime is only linear in \( \mathcal{G}(x_t) \), which may be substantially smaller than \( |G| \). For example, this property holds for collections \( \mathcal{G} \) of groups induced by conjunctions or disjunctions of binary features. Finally, we observe that our runtime is entirely independent of the choice of the discretization parameter \( r \).

5 Online Multivalid Marginal Coverage

In this section, we derive an online algorithm which supplies multivalid prediction intervals satisfying a given coverage target \( 1 - \delta \). We follow the same basic strategy that we developed in Section 4 for making multicalibrated mean predictions, with a couple of important deviations.

First (as described in Section 3), in order for prediction intervals with given target coverage to even exist, we need to assume a constrained adversary who plays smooth distributions. As a result, our minimax theorem-based existential derivation of a multivalid algorithm has to handle this new, more complex strategy space for the Adversary.

Second, when it comes to instantiating our existential multivalid algorithm, we face the added difficulty that unlike in the case of mean multicalibration, the Learner’s minimax strategies do not appear to have any nice structure. As a result, in order to efficiently obtain the Learner’s strategy at each round, we have to come up with an efficient way to solve an exponentially large linear program that defines this minimax strategy. Specifically, we show how to solve this LP in polynomial time via the Ellipsoid algorithm using a simple greedy separation oracle.

5.1 An Existential Derivation of the Algorithm and Multicoverage Bounds

Our goal in this section is to derive an algorithm which at each round, makes predictions \((\ell_t, u_t) \in \mathcal{P}_{\text{interval}}\) that are multivalid with respect to some target coverage probability \( 1 - \delta \).

Towards this end, we define the coverage error of a group \( G \) and interval \((\ell, u)\):

\[
V_{G, (\ell, u)}^{G} = \sum_{t=1}^{s} I[x_t \in G, (\ell_t, u_t) \in B_n(i, j)] \cdot v_\delta((\ell_t, u_t), y_t),
\]

where \( v_\delta((\ell, u), y) = \text{Cover}((\ell, u), y) - (1 - \delta) \).

Just as before, our coverage error serves as a bound on our multicoverage error.

\textbf{Definition 18.} Given a transcript \( \pi = (x_t, (\ell_t, u_t), y_t)_{t=1}^{s} \), we define the coverage error for a group \( G \in \mathcal{G} \) and bucket \((i, j) \in [n] \times [n] \) at time \( s \) to be:

\[
V_{G, (\ell, u)}^{G} = \sum_{t=1}^{s} I[x_t \in G, (\ell_t, u_t) \in B_n(i, j)] \cdot v_\delta((\ell_t, u_t), y_t),
\]

where \( v_\delta((\ell, u), y) = \text{Cover}((\ell, u), y) - (1 - \delta) \).

Observation 5.1. Fix a transcript \( \pi_T \). If for all \( G \in \mathcal{G} \), and buckets \((i, j) \in [n] \times [n] \),

\[
\left| V_{T, (\ell, u)}^{G} \right| \leq \alpha T
\]

then the corresponding sequence of prediction intervals are \((\alpha, n)\)-multivalid with respect to \( \mathcal{G} \).
To bound the maximum absolute value of our coverage errors across all groups and interval predictions, we again introduce the same style of surrogate loss function:

**Definition 19 (Surrogate loss).** Fixing a transcript $\pi_s \in \Pi^*$ and a parameter $\eta \in (0, 1/2)$, define a surrogate coverage loss function at day $s$ as:

$$L_s(\pi_s) \equiv \sum_{G \in G_V(i,j) \in [n] \times [n]} \left( \exp(\eta V_s^G(i,j)) + \exp(-\eta V_s^{G,G(i,j)}) \right),$$

where $V_s^{G,G(i,j)}$ are implicitly functions of $\pi_s$. When the transcript is clear from context we will sometimes simply write $L_s$.

Once again, $0 < \eta < 0$ is a parameter that we will set later.

As before, we proceed by bounding the conditional change in the surrogate loss function. For fixed $\pi_s \in \Pi^*$, $x_{s+1} \in \mathcal{X}$ and an interval $(\ell, u) \in \mathcal{P}_{\text{interval}}$, we define it as

$$\Delta_{s+1}(\pi_s, x_{s+1}, (\bar{\ell}_s, \bar{u}_s)) \equiv \mathbb{E}_{\bar{y}_{s+1}}[\bar{L}_{s+1} - L_s(x_{s+1}, (\bar{\ell}_s, \bar{u}_s), \pi_s)]$$

Before stating our bound on this conditional change, we define, for convenience, the quantities

$$C_{s, \ell, u}(x_{s+1}) \equiv \sum_{G \in G_V(x_{s+1})} \exp(\eta V_s^{G(i,j)}) - \exp(-\eta V_s^{G(i,j)})$$

for each $i \leq j \in [n]$. When $x_{s+1}$ is clear from context, we will elide it and simply write $C_{s,i,j}$.

**Lemma 20.** For every transcript $\pi_s \in \Pi^*$, $x_{s+1} \in \mathcal{X}$, and $(\bar{\ell}_s, \bar{u}_s) \in \mathcal{B}_n(i, j)$, we have:

$$\Delta_{s+1}(\pi_s, x_{s+1}, (\bar{\ell}_s, \bar{u}_s)) \leq \left( \eta \mathbb{E}_{\bar{y}_{s+1}}[\bar{v}_{s,i,j}((\bar{\ell}_s, \bar{u}_s), \bar{y}_{s+1})] \right) C_{s,i,j}(x_{s+1}) + 2\eta^2 L_s$$

Next, we abuse notation and write $V_s^{G,\ell,u}$ to denote $V_s^{G(i,j)}$ for $i, j \in [n] \times [n]$ such that $(\ell, u) \in \mathcal{B}_n(i, j)$. Similarly, given $(\ell, u) \in \mathcal{B}_n(i, j)$, we let $C_{s,u} \equiv C_{s,i,j}$. That is, fixing $\pi_s$ and $x_{s+1}$, for any $(\ell, u) \in \mathcal{P}_{\text{interval}}$ such that $(\ell, u) \in \mathcal{B}_n(i, j)$,

$$C_{s,\ell, u}(x_{s+1}) \equiv C_{s, i,j}(x_{s+1}) = \sum_{G \in G_V(x_{s+1})} \exp(\eta V_s^{G(i,j)}) - \exp(-\eta V_s^{G(i,j)}).$$

**Definition 21.** We write $Q_{\rho, r_n}$ for the set of all $(\rho, r_n)$ smooth distributions over $[0, 1]$. We write $\hat{Q}_{\rho, r_n}$ for the set of all $(\rho, r_n)$-smooth distributions whose support belongs to the grid $\mathcal{P}_{\text{interval}} = \{0, \frac{1}{r_n}, \ldots, 1\}$:

$$\hat{Q}_{\rho, r_n} \equiv \Delta \mathcal{P}_{\text{interval}} \cap Q_{\rho, r_n}.$$

We will show (in Claim 23) that when the learner is restricted to selecting intervals from $\mathcal{P}_{\text{interval}}$ without loss of generality, rather than considering adversaries that play arbitrary distributions over $Q_{\rho, r_n}$, it suffices to consider adversaries that play discrete distributions from $\hat{Q}_{\rho, r_n}$, which will be more convenient for us.

**Lemma 22 (Value of the Game).** For any $x_{s+1} \in \mathcal{X}$, any adversary restricted to playing $(\rho, r_n)$-smooth distributions, and any transcript $\pi_s \in \Pi^*$, there exists a distribution over predictions for the learner $Q_s^{\ell, u} \in \Delta \mathcal{P}_{\text{interval}}$ which guarantees that:

$$\mathbb{E}_{(\ell, \pi) \sim Q_{s+1}^{\ell, u}}[\Delta_{s+1}(\pi_s, x_{s+1}, (\bar{\ell}_s, \bar{u}_s))] \leq L_s(\eta \rho + 2\eta^2).$$
Proof. We again proceed by defining a zero-sum game with objective function equal to the upper bound on $\Delta_{s+1}(\pi_s, x_{s+1}, (\tilde{t}_{s+1}, \tilde{u}_{s+1}))$ that we proved in Lemma 20:

$$u((\ell, u), y) = \eta \cdot v_i((\ell, u), y) \cdot C_{s,u} + 2\eta^2 L_s.$$ 

Here, the strategy space for the learner (the minimization player) is the set of all distributions over $\mathcal{P}^n_{\text{interval}}$: $Q^L = \Delta \mathcal{P}^n_{\text{interval}}$. A priori, the strategy space for the adversary is $Q, \rho, \eta$ – the set of all $(\rho, \eta)$-smooth distributions, but we show that taking $Q^A = \hat{Q}, \rho, \eta$, the set of all discrete $(\rho, \eta)$-smooth distributions, yields the same value of the game.

Claim 23. For any $Q^L \in \Delta \mathcal{P}^n_{\text{interval}}$ for the learner, the adversary has a best response amongst the set of all $(\rho, \eta)$-smooth distributions with support only over the discretization $\{0, 1, \ldots, \eta n\}$. In other words, for any $Q^L \in \Delta \mathcal{P}^n_{\text{interval}}$, there exists $Q^A \in \hat{Q}, \rho, \eta$ such that:

$$\hat{Q}^A \in \arg \max_{Q^A \in \hat{Q}, \rho, \eta} \mathbb{E}_{\ell,u} \left[ u((\ell, u), y) \right].$$

Proof. Fix any $Q^{A'} \in \arg \max_{Q^A \in \hat{Q}, \rho, \eta} \mathbb{E}_{\ell,u} \left[ u((\ell, u), y) \right]$, an arbitrary $(\rho, \eta)$-smooth best response for the maximization player. It is easy to check that we can define a discrete $(\rho, \eta)$-smooth $Q^A \in \hat{Q}, \rho, \eta$, that obtains the same objective value as $Q^{A'}$, as follows:

$$\Pr_{y \sim Q^{A'}} [y = \frac{i}{\eta n}] = \Pr_{y \sim Q^{A'}} [y \in [\frac{i-1}{\eta n}, \frac{i}{\eta n}]], \text{ for all } i \in \{0\} \cup [\eta n].$$

By Observation 3.1, for any $(\rho, \eta)$-smooth label distribution $Q^A$, there exists an interval $(\ell, u) \in \mathcal{P}^n_{\text{interval}}$ such that $\Pr_{y \sim Q^A} [y \in [\ell, u] - (1 - \delta)] \leq \rho$ – that is, there exists $(\tilde{t}, \tilde{u})$ with $\mathbb{E}_{g_{s+1}} [v_s((\tilde{t}, \tilde{u}), \tilde{y}_{s+1})] \leq \rho$. We can thus bound the value of our game as follows:

$$\max_{Q^A \in \hat{Q}, \rho, \eta} \min_{(\ell, u) \in \mathcal{P}^n_{\text{interval}}} \mathbb{E}_{y \sim Q^A} \left[ u((\ell, u), y) \right] \leq \sum_{g(x_{s+1})} \exp(\eta V^G_s((\ell, u))) (\eta \rho) + \exp(-\eta V^G_s((\ell, u))) (\eta \rho) + 2\eta^2 L_s,$$

$$\leq L_s (\eta \rho + 2\eta^2).$$

It is easy to verify that $\Delta \mathcal{P}^n_{\text{interval}}$ and $\hat{Q}, \rho, \eta$ are both compact and convex. The lemma then follows by applying the minimax theorem (Theorem 5).

Corollary 24. For every $s \in [T]$, $\pi_s \in \Pi^*$, and $x_{s+1} \in \mathcal{X}$ (which fixes $L_s$ and $Q^L_{s+1}$), and any distribution over $\mathcal{X}$:

$$\mathbb{E}_{(\ell,u) \sim Q^L_{s+1}} [\tilde{L}_{s+1} | \pi_s] \leq L_s + \mathbb{E}_{(\ell,u) \sim Q^L_{s+1}} [\Delta_{s+1}(\pi_s, x_{s+1}, (\tilde{t}_{s+1}, \tilde{u}_{s+1}))] \leq L_s (1 + \eta \rho + 2\eta^2).$$

As with mean multicalibration, Lemma 22 defines (existentially) an algorithm that the learner can use to make predictions – Algorithm 3.

Algorithm 3 A Generic Multi-valid Predictor.

\begin{enumerate}
\item for $t = 1, \ldots, T$ do
\begin{enumerate}
\item Observe $x_t$. Given $\pi_{t-1}$ and $x_t$, let $Q^L_t \in \Delta \mathcal{P}^n_{\text{interval}}$ be the distribution over prediction intervals whose existence is established in Lemma 22.
\item Sample $(\tilde{t}, \tilde{u}) \sim Q^L_t$ and predict $(\tilde{t}, \tilde{u}) = (\tilde{t}, \tilde{u})$
\end{enumerate}
\end{enumerate}
Using similar techniques as for mean multicalibration, one can now show that Algorithm 3 (if one could compute the distributions $Q^L_t$) results in multivalid (in expectation and with high probability) prediction intervals. This proves the existential Theorem 8 stated in Section 3. We do not include the derivation for the sake of brevity, but as a quick illustration of its similarity to the mean-multicalibration derivation, observe that Algorithm 3 results in the following surrogate loss bound:

$$E_{x_t} [\tilde{L}_T] \leq 2|G|n^2 \exp (T\eta \rho + 2T\eta^2).$$

This is obtained via Corollary 24 and using the first part of Theorem 15 (Bound (2)) with $L_0 = 2|G|n^2$, $c = \rho$. From here, the in-expectation multivalidity bound is within easy reach.

5.2 Deriving an Efficient Algorithm via Equilibrium Computation

We now show how to implement Algorithm 3 via efficiently sampling, at each round $t$, from the Learner’s minimax distribution $Q^L_t$ whose existence we established in Lemma 22. It is easy to observe that $Q^L_t$ can be computed by solving the following linear program:

$$\min_{Q^L \in \mathcal{P}_{\text{interval}}} \gamma \text{ s.t.}$$

$$\forall Q^A \in \hat{Q}_{\rho,rn} : \sum_{y \in \mathcal{P}_{\text{interval}}} Q^A(y) \left( \sum_{(l,u) \in \mathcal{P}_{\text{interval}}} Q^L((l,u)) \left( v(l,(l,u) C_{t-1}^l(x_t) \right) \right) \leq \gamma,$$

$$\sum_{(l,u) \in \mathcal{P}_{\text{interval}}} Q^L((l,u)) = 1,$$

$$\forall (l,u) \in \mathcal{P}_{\text{interval}} : Q^L((l,u)) \geq 0.$$

Figure 1 Linear Program for Computing the Learner’s Minimax Strategy at Round $t$.

However, a priori, it is unclear how to solve this LP in polynomial time, as it has $(rn)^2 + 1$ variables and an infinite number of constraints. However, as we will show, the number of constraints can in fact be taken to be finite (albeit exponentially large), and we have an efficient separation oracle to identify violated constraints. Together, these two facts allow us to apply the Ellipsoid algorithm to efficiently solve LP 1, thus obtaining $Q^L_t$.

Algorithm 4 Von Neumann’s Multivalid Predictor.

INPUT: $\epsilon > 0$.

for $t = 1, \ldots, T$ do

Observe $x_t$ and compute $C_{t-1}^l(x_t)$ for each $(l,u) \in \mathcal{P}_{\text{interval}}$ as in (5).

Solve the Linear Program from Figure 1 using the Ellipsoid algorithm, with Algorithm 5 as a separation oracle, to obtain an $\epsilon$-approximate solution $Q^L_t \in \mathcal{D}_{\mathcal{P}_{\text{interval}}}.$

Predict $(\tilde{l}_t, \tilde{u}_t) = (l,u)$ with probability $Q^L_t((l,u))$.

Theorem 25. Algorithm 4 implements Algorithm 3. In particular, it obtains multivalidity guarantees arbitrarily close to those of Theorem 8. Namely, for any desired $\epsilon > 0$, Algorithm 4 obtains the following in-expectation and high-probability multivalidity bounds:

$$E_{x_t} [\tilde{L}_T] \leq 2|G|n^2 \exp (T\eta \rho + 2T\eta^2).$$

20 By choosing, respectively, $\eta = \sqrt{\frac{\ln(2|G|n^2 + \epsilon)}{2T}}$ or $\eta = \sqrt{\frac{\ln(2|G|n^2 + \epsilon)}{2T}}$. 

We can write this as a linear program, over the space $\mathcal{G}$. Although in fact, in the proof of Lemma 27, we will show that without loss of generality we can equivalently impose only finitely (but exponentially) many constraints.

Recall that at each round $t$, we need to find an equilibrium game strategy for the learner in the zero-sum game defined by the objective function:

$$u((\ell, u), y) = \eta v_\delta((\ell, u), y) C_{t-1}^{\ell, u} + 2\eta^2 L_{t-1}$$

In this game, the strategy space for the learner is the set of all distributions over discrete intervals: $Q^L = \Delta P_{\text{interval}}^n$, and (by Lemma 23), the action space for the adversary can be taken to be the set of all discrete smooth distributions: $Q^A = \hat{Q}_{p, rn}$.

The equilibrium structure of a game is invariant to adding and multiplying the objective function by a constant. Hence we can proceed to solve the game with the objective function:

$$u((\ell, u), y) = (\text{Cover}((\ell, u), y) - (1 - \delta)) C_{t-1}^{\ell, u}.$$  

To compute an equilibrium of the game, we need to solve for a distribution $Q^L$ satisfying:

$$Q^L \in \arg\min_{Q^L \in \Delta P_{\text{interval}}^n} \max_{Q^A \in \hat{Q}_{p, rn}} \mathbb{E}_{y \sim Q^A((\ell, u)) \sim Q^L} [u((\ell, u), y)].$$

We can write this as a linear program, over the $O((rn)^2)$ variables $Q^L((\ell, u))$: see Figure 1. A priori, this linear program has infinitely many constraints. Nevertheless, we show that we can efficiently implement a separation oracle, which given a candidate solution $(\gamma, \gamma)$, can find a violated constraint whenever one exists. This is sufficient to efficiently find, using the Ellipsoid algorithm, a feasible solution of the linear program achieving value within any desired $\epsilon > 0$ of the optimum.

> **Claim 27.** Algorithm 5 is a separation oracle for LP 1. It runs in time $O((rn)^3)$.

Proof. Given $Q^L \in \Delta P_{\text{interval}}^n$, $\gamma \in \mathbb{R}$, the oracle determines if there is $Q^A \in \hat{Q}_{p, rn}$ such that:

$$\sum_{y \in \hat{Q}_{p, rn}} Q^A(y) \left( \sum_{(\ell, u) \in \hat{Q}_{p, rn}} Q^L((\ell, u)) \left( v_\delta((\ell, u), y) C_{t-1}^{\ell, u} \right) \right) > \gamma.$$  

Recall $v_\delta((\ell, u), y) = \text{Cover}((\ell, u), y) - (1 - \delta)$. Fixing the learner’s choice of $Q^L \in \Delta P_{\text{interval}}^n$, those terms in this objective that involve $(1 - \delta)$, are independent of the choice of $Q^A \in \hat{Q}_{p, rn}$. Hence, we may take them out and let the adversary choose $Q^A$ to equivalently maximize:

$$\tilde{u}(Q^L, Q^A) = \sum_{i \in \{0, \ldots, rn\}} Q^A \left( \frac{i}{rn} \right) \sum_{(\ell, u) \in \hat{Q}_{p, rn}, \text{Cover}((\ell, u), \frac{i}{rn}) = 1} Q^L((\ell, u)) C_{t-1}^{\ell, u} = \sum_{i \in \{0, \ldots, rn\}} Q^A \left( \frac{i}{rn} \right) W_i.$$  

\[21\] Although in fact, in the proof of Lemma 27, we will show that without loss of generality we can equivalently impose only finitely (but exponentially) many constraints.
Algorithm 5 A Separation Oracle for Linear Program 1.

INPUT: A proposed solution $Q^L, \gamma$ for Linear Program 1
OUTPUT: A violated constraint of LP 1 if one exists, or a certification of feasibility.

for $i = 0, 1, \ldots, rn$ do
Compute $W_i \equiv \sum_{(\ell,u) \in P^{rn}_{\text{interval}} : \text{Cover}((\ell,u),\frac{1}{\gamma}) = 1} Q^L((\ell,u)) C_{t-1}^{\ell,u}$.

Let $\sigma : \{0, \ldots, rn\} \rightarrow \{0, \ldots, rn\}$ be a permutation s.t. $W_{\sigma(0)} \geq W_{\sigma(1)} \geq \ldots \geq W_{\sigma(rn)}$.

for $i = 0, 1, \ldots, rn$ do
Set $Q^A(\sigma(i)) = \min(\rho, 1 - \sum_{j=0}^{i-1} Q^A(\sigma(j))$.

if $\sum_{y \in P^{rn}} Q^A(y) \left(\sum_{(\ell,u) \in P^{rn}_{\text{interval}}} Q^L((\ell,u)) \left(v_{\delta}(\ell,u,y) C_{t-1}^{\ell,u}\right)\right) > \gamma$, or $Q^L$ not a prob. dist. then
return the violated constraint.

return FEASIBLE

Finding $Q^A \in \hat{Q}_{\rho,rn}$ that maximizes this expression is a fractional knapsack problem: each “item” $i \in \{0, \ldots, rn\}$ has value $W_i$ and quantity $\rho$ (as $Q^A$ is smooth, $Q^A(\frac{1}{\gamma}) \leq \rho$), and the knapsack capacity is 1. This is precisely the problem solved by Algorithm 5. ▷

Finally, showing the claimed multivalidity guarantees (as a function of $\epsilon > 0$) amounts to repeating several calculations from the proof of the existential result with an $\epsilon$ error term. ▶

References