When Does a Predictor Know Its Own Loss?

Aravind Gollakota[1]
Apple, Cupertino, CA, USA

Parikshit Gopalan
Apple, Cupertino, CA, USA

Aayush Karan[2]
Harvard University, Cambridge, MA, USA

Charlotte Peale[2]
Stanford University, Stanford, CA, USA

Udi Wieder
Apple, Cupertino, CA, USA

Abstract

Given a predictor and a loss function, how well can we predict the loss that the predictor will incur on an input? This is the problem of loss prediction, a key computational task associated with uncertainty estimation for a predictor. In a classification setting, a predictor will typically predict a distribution over labels and hence have its own estimate of the loss that it will incur, given by the entropy of the predicted distribution. Should we trust this estimate? In other words, when does the predictor know what it knows and what it does not know?

In this work we study the theoretical foundations of loss prediction. Our main contribution is to establish tight connections between nontrivial loss prediction and certain forms of multicalibration [20], a multigroup fairness notion that asks for calibrated predictions across computationally identifiable subgroups. Formally, we show that a loss predictor that is able to improve on the self-estimate of a predictor yields a witness to a failure of multicalibration, and vice versa. This has the implication that nontrivial loss prediction is in effect no easier or harder than auditing for multicalibration. We support our theoretical results with experiments that show a robust positive correlation between the multicalibration error of a predictor and the efficacy of training a loss predictor.

2012 ACM Subject Classification Theory of computation → Machine learning theory

Keywords and phrases loss prediction, multicalibration, active learning, algorithmic fairness, calibration, predictive uncertainty, uncertainty estimation, machine learning theory

Digital Object Identifier 10.4230/LIPIcs.FORC.2025.22

Related Version *Full Version*: `https://arxiv.org/abs/2502.20375`

Funding *Aayush Karan*: Supported by the Paul and Daisy Soros Fellowship for New Americans. *Charlotte Peale*: Supported by the Apple Scholars in AI/ML PhD fellowship and the Simons Foundation Collaboration on the Theory of Algorithmic Fairness.

Acknowledgements We thank Moises Goldszmidt, Shayne O'Brien, Daniel Tsai, Robert Fisher and Dor Shaviv for introducing us to this problem and for sharing their insights on and experience with practical loss prediction and its applications.

1 Introduction

It is increasingly common for large machine learning models to be part of a pipeline where a base model is trained by a provider that has access to large-scale data and computational power, and the model is then deployed by a heterogeneous set of downstream consumers, for

[1] Authors in alphabetical order.
[2] Work done while interning at Apple.

a diverse range of prediction tasks. Not only could the tasks be very different from each other, they might involve data distributions, loss functions and other metrics and features that are markedly different from those used to train the model. Indeed, often the data sources used in training are not even disclosed to the downstream application. A typical instantiation of this framework is zero-shot classification, where (say) an LLM is required to classify texts into classes described by the user. Another important case is that of medical classification where the base model was trained on one set of features, say lab reports, but the model (or human) downstream has access to additional features such as patient history.

In such a situation, a user might want to delve deeper into how the model is likely to perform on their specific task. They might seek to discover problematic regions of the input space where the model performs poorly, where performance is measured by an appropriate loss function chosen by the user. This information could prove valuable in several ways:

- Active and continual learning: Users could address performance issues by collecting additional data points from problematic regions and fine-tune the model on this enhanced dataset.
- Fairness considerations: The analysis might reveal potential biases or inequities in the model's performance across different subgroups.
- Selective prediction: Such insights could guide downstream users on when to rely on the model's predictions and when to exercise caution. In cases where predictions are likely to be unreliable, users might opt to consult external experts or alternative models instead.

By systematically identifying and addressing these performance vulnerabilities, users can judge the model's reliability, fairness, and overall utility. A provider who desires to improve their model would similarly benefit from knowing where their model performs poorly.

This discussion motivates the problem of loss prediction, which we now define. In this work we focus for concreteness on the binary classification setting, although many of the results extend to multiclass classification as well. We are given a pre-trained predictor $p : \mathcal{X} \to [0,1]$ (where $\mathcal{X}$ denotes the space of inputs), a target loss $\ell : \{0,1\} \times [0,1] \to \mathbb{R}$[3], and some labeled data (x, y) drawn from an unknown distribution $\mathcal{D}$ on $\mathcal{X} \times \{0,1\}$. The goal is to estimate the loss $\ell(y, p(x))$ incurred at a point x using a loss predictor $\mathsf{LP}_{\ell,p} : \mathcal{X} \to \mathbb{R}$. This can be viewed as a regression problem, and we measure the quality of a loss predictor by its expected squared loss with respect to the true loss, i.e. $\mathbb{E}_{(x,y)\sim\mathcal{D}}[\big(\mathsf{LP}_{\ell,p}(x) - \ell(y, p(x))\big)^2]$.

Loss prediction is closely connected to the well-studied problem of uncertainty estimation. A standard measure of predictive uncertainty at a point is the expected loss that a predictor suffers at that point [30], and estimating this requires solving the problem of loss prediction. Given such a loss predictor, its uncertainty estimate is then often decomposed into two parts: aleatoric uncertainty, which is the uncertainty stemming from the randomness in nature, and epistemic uncertainty, which is the uncertainty arising from shortcomings in our model and/or training data.[4] Since epistemic uncertainty can be driven down with more data and fine tuning, active learning strategies have been proposed that use loss predictors to decide what regions to prioritize for collecting more data [41, 31]. Loss predictions can also be used for various other applications, including deciding when a model should abstain from making

[3] It will be convenient to assume that this loss is *proper*, namely that for any p^* the expected true loss $\mathbb{E}_{y\sim p^*}[\ell(y,p)]$ is minimized at $p = p^*$. Canonical examples are the cross-entropy and the squared loss. The case of general losses can be reduced to that of proper losses.

[4] There are many proposals for how to achieve such a decomposition, see e.g. [23], not all of which come with rigorous guarantees. Recent work of [2] does give rigorous guarantees, but it requires an enhancement to the standard learning model called learning with snapshots. See also the discussion of related work therein.

a prediction or route the input to a stronger model. Consequently, there has been plenty of applied work on the problem of loss prediction, but little theoretical analysis (see Section 7 for more discussion of related work).

1.1 Our contributions

In this work, we initiate a study of the theoretical foundations of loss prediction. We formalize the task of loss prediction and connect it to the basic primitives of computational learning.

The self-entropy predictor of loss

The first question is what baseline one should use to measure the quality of a loss predictor. Drawing from work on outcome indistinguishability [7, 9], we propose a baseline based on the fact that a predictor posits a certain model of nature: that labels for x are drawn according to a Bernoulli distribution with parameter $p(x)$. This entails a belief about the expected loss it will incur at a point, which is $\mathbb{E}_{\tilde{y}\sim\mathrm{Ber}(p(x))}[\ell(\tilde{y}, p(x))]$. In the case of squared loss, this estimate is $p(x)(1-p(x))$ at the point x; for the cross-entropy loss, it is the Shannon entropy $H(p(x))$. By results of [13], for any proper loss ℓ, there exists a concave "generalized entropy" function $H_\ell : [0,1] \to \mathbb{R}$ such that this estimate is $H_\ell(p(x))$. We refer to this as the self-entropy predictor. Using this as our baseline, we ask when it is possible for a loss predictor to do better than the self-entropy predictor. At a high level, we wish to understand

When can a loss-predictor beat a model in estimating what the model knows and does not know?

A hierarchy of loss prediction models

It is natural that loss predictors should receive the input features $i(x)$[5] and the prediction $p(x)$ as inputs. But this does not capture some important architectures for loss prediction that are used in practice; for instance the works of [41, 28] which consider models that can access representations of x that are computed by the neural network computing p. Accordingly, we model loss predictors as taking inputs $\varphi(p, x)$ lying in an abstract feature space and returning a loss prediction $\mathsf{LP}_{\ell,p}(p(x))$. We define a hierarchy of loss predictors of increasing strength, depending on expressivity of $\varphi(p, x)$ (Definition 2):

1. *Prediction-only loss predictors* only have access to p's prediction at a point x, i.e. $\varphi(p, x) = p(x)$. The self-entropy predictor of loss is an example.
2. *Input-aware loss predictors* have additional access to the input features $i(x)$, i.e. $\varphi(p, x) = (p(x), i(x))$.
3. *Representation-aware loss predictors* have access to $\varphi(p, x) = (p(x), i(x), r(x))$, where $r(x)$ is some representation of x. In this case, we further distinguish between two settings:
   - Internal representations $r(x) = r_p(x)$ where $r_p(x)$ is computed by p in the course of computing $p(x)$.
   - External representations $r(x) = r_e(x)$ which are not explicitly computed by p.

Internal representations could for instance correspond to the embedding produced by the last few layers of a deep neural net. External representations could be the representation of x obtained from a different model, or additional features added by consulting human experts. The related work in Section 7 gives examples of both kinds of representations that have been considered in the literature.

[5] For clarity, we make a distinction between the abstract input x (e.g., an individual) and its input feature representation $i(x)$ (e.g., features collected about the individual).

Finally, we define the advantage of a loss predictor over the self-entropy predictor to be the difference in the squared loss incurred by the two loss predictors (Definition 4).

Relation to auditing for multicalibration

Multicalibration is a multigroup fairness notion introduced by [20], which has since found numerous other applications [27, 7, 16]. We show that learning a loss predictor with a non-trivial advantage is tightly connected to auditing the predictor for multicalibration. At a high level, we show the following correspondence, which we formalize in Theorem 7:

Finding a **prediction-only** loss predictor with good advantage	$\Leftrightarrow$	Identifying a **calibration** violation for p
Finding an **input-aware** loss predictor with good advantage	$\Leftrightarrow$	Identifying a **multicalibration** violation for p
Finding a **representation-aware** loss predictor with good advantage	$\Leftrightarrow$	Identifying a **representation-aware multicalibration** violation for p, where the auditor function is of the form $c(\varphi(p, x))$.

In all cases, the regions where the multicalibration violations occurs arise from analyzing where the loss predictor and the self-entropy predictor differ from each other. The first two notions in our hierarchy, calibration and multicalibration, have been extensively studied in previous works [11, 20]. The last member of the hierarchy, representation-aware multicalibration, is a strengthening of multicalibration that naturally extends the multicalibration framework.

Furthermore, we explore how the lens of multicalibration proves valuable in predicting well for a large class of losses, particularly when learning individual predictors for each loss is impractical. In Theorem 14, we show that via standard techniques for learning multicalibrated predictors, we can efficiently learn a predictor whose self-entropy predictions for every 1-Lipschitz proper loss (of which there are infinitely many) are comparable to the best-in-class loss predictor for each loss from some fixed class of candidate predictors.

On calibration blind-spots for loss prediction

Calibration is not necessary for producing good estimates of the true loss. For instance, a predictor that predicts $p(x) = 1/2$ on every input will indeed incur a squared loss of $1/4$, matching its self-entropy predictor regardless of the true labels. But depending on the distribution of labels, this predictor might be very far from calibrated, and need not even be accurate in expectation.

Our results imply a simple characterization of such "blind spots" for any proper loss ℓ as points p where $H'_\ell(p) = 0$.[6] In terms of the loss ℓ, this is equivalent to $\ell(0, p) = \ell(1, p)$, so that the loss incurred is independent of the label, and hence predicting the expected loss for such p is trivial. For strictly proper losses, the function H_ℓ is strictly concave, and there is a unique point where this happens.

[6] Recall that $H_\ell(p)$ is the concave entropy function corresponding to ℓ.

This introduces some subtlety in the type of multicalibration violations that arise from our correspondence; the standard calibration error at p is weighted by a factor of $H'_\ell(p)$. Hence non-trivial loss prediction corresponds to (multi)calibration violations at prediction values p such that $H'_\ell(p)$ is far from 0.

Experimental results

We empirically verify that there is a correspondence between loss prediction and multicalibration (see Section 6). Focusing on input-aware loss prediction algorithms run across a variety of base predictor types, we find that:

- As the multicalibration error of the base model increases, the advantage of the loss prediction over the self estimate of the loss increases.
- Loss predictors are more advantageous on data subgroups that have higher calibration error.

Our experiments suggest that regression-based loss predictors present an effective way to audit for multicalibration and are an intriguing avenue towards developing efficient multicalibration algorithms for practice.

1.2 Takeaways from our result

The main takeaway from our work is that non-trivial loss prediction is no easier (and not much harder) than auditing the predictor itself. Any predictor that improves over the self-entropy predictor could be used to find (and possibly fix) multicalibration issues in the predictor.

Practical multicalibration using loss prediction

The complexity of multicalibration depends crucially on the class of test functions used. For complex functions, our equivalence suggests a novel approach to multicalibration auditing: choose a proper loss, run a regression for loss prediction, and see if the loss predictor outperforms the self-entropy predictor. This is a simple and practical approach that is able to leverage the strength of any well-engineered library for regression. In our experiments we show that this is indeed effective, with loss prediction advantage being robustly correlated with multicalibration error across mutliple base predictors as well as subgroups.[7]

On two-headed architectures

There has been work on training deep neural nets with two heads: a prediction head p, and a loss prediction head $\mathsf{LP}_{\ell,p}$ [41, 28]. The loss prediction head has access to the embedding of the inputs produced by the last few layers of the neural net, and can be modeled by a representation-aware loss predictor of low complexity. Our result shows that (at least in a classification setting) one of the following must be true:

- The loss prediction head does not give much advantage over the self-entropy predictor, which only requires prediction access.

[7] To get around the blind-spot issue, one could choose a few strictly proper loss functions each with a different blind spot. This is easy to do, given the correspondence between convex functions and proper losses [13].

- The prediction head is not optimal, as evidenced by a multicalibration violation witnessed by the difference between the loss prediction head and the self-entropy predictor (see Lemma 10).

The ideal situation for a well-trained model is clearly the former.

Note that this does not mean that two-headed architectures are not useful: the two heads may influence the training dynamics in a subtle way, with the loss-predictor head revealing complex regions where multicalibration fails. However, what our result implies is that when training concludes, we want to be in the situation where the loss-predictor is not much better than the self-entropy predictor. This is analogous to the situation with GANs [14], where at the end of training, we would ideally like the generator to be able to fool the discriminator. But in the intermediate stages of training, the discriminator helps improve the quality of the generator.

Two-headed architectures may also be useful in prediction problems more general than ordinary classification, such as image segmentation [28], where a predictor does not necessarily come with a self-entropy estimate at all.

Extra information helps: when loss prediction might be effective

An important scenario is where the loss-predictor may have informative features $\varphi'(x)$ about the input x that were not available to the entity that was training the model p. For example, consider a neural net that is trained to screen X-rays for prevalence of a certain medical condition. Such models may be trained by aggregating data from across several hospitals. A hospital that is trying to use this model might not have the same computational resources available to them. But they might have access to other useful information such as observations made by a doctor or the patient's medical history.

In such a case, even a model which is multicalibrated with respect to complex functions over the features $\varphi(p, x)$ might not be multicalibrated with respect to simple functions over a new set of features. This was illustrated in the recent work of [3] in their work on incorporating human judgments to improve on model predictions. Another natural scenario in which the loss predictor may have extra information is if it uses a powerful pretrained foundation model. The work of [24] does precisely this, leveraging embeddings from CLIP [35]. In such settings, improvements to the predictor, and loss predictors with a non-trivial advantage are both possible.

Organization

We define loss predictors in Section 2 and recall the relevant notions of multicalibration in Section 3. We present the equivalence between loss prediction with an advantage over the self-entropy predictor and multicalibration in Section 4. We discuss how to efficiently find predictors that give good self-entropy predictors for multiple loss functions in Section 5. In Section 6, we empirically demonstrate the correspondence between loss prediction advantage and multicalibration violations, and show that it holds across multiple architectures and data subgroups. We discuss related work in detail in Section 7. In Appendix A, we present the extension of our results to the case where the losses are non-proper.

2 Loss prediction

We consider binary classification, with a distribution $\mathcal{D}$ on $\mathcal{X} \times \{0, 1\}$. A predictor is a function $p : \mathcal{X} \to [0, 1]$. The Bayes optimal predictor is defined as $p^*(x) = \mathbb{E}[y|x]$. Given p, we define the simulated distribution $\mathcal{D}(p)$ on $\mathcal{X} \times \{0, 1\}$ where x is drawn as in $\mathcal{D}$, and

$y|x \sim \mathrm{Ber}(p(x))$. Let $\ell : \{0,1\} \times [0,1] \to [0,1]$ be a proper loss function.[8] We will use the following characterization of proper losses.

▶ **Lemma 1** ([13]). *For every proper loss ℓ, there exists a concave function $H_\ell : [0,1] \to \mathbb{R}$ so that*

$$\ell(y, v) = H_\ell(v) + (y - v)H'_\ell(v).$$

where $H'_\ell(v)$ is a "superderivative" of H_ℓ, i.e. for any $v, w \in [0,1]$, $H_\ell(v) \leq H_\ell(w) + (v - w)H'_\ell(w)$.

When $H_\ell(v)$ is differentiable at all $v \in [0,1]$, the superderivative is unique, and is just the derivative. From the definition it follows that

$$H_\ell(v) = \mathop{\mathbb{E}}_{y \sim \mathrm{Ber}(v)}[\ell(y, v)] \in [0,1]$$
$$H'_\ell(v) = \ell(1, v) - \ell(0, v) \in [-1, 1]$$

Let $L(p^*; p) = \mathbb{E}_{y \sim \mathrm{Ber}(p^*)}[\ell(y, p)]$ denote the expected loss when $y \sim \mathrm{Ber}(p^*)$ but we predict p. Then

$$L(p^*; p) = H_\ell(p) + (p^* - p)H'_\ell(p) \geq H_\ell(p^*) = L(p^*; p^*) \tag{1}$$

where the inequality follows from the concavity of H_ℓ, and is equivalent to the loss ℓ being proper.

We now define the notion of a loss predictor.

▶ **Definition 2** (Loss predictor). *Let p be a predictor and ℓ be a proper loss. Let Φ be an abstract feature space, which we will make concrete shortly. A* loss predictor *is a function $\mathsf{LP}_{\ell,p} : \Phi \to \mathbb{R}$, which takes as input some features $\varphi(p, x) \in \Phi$ related to a point x and its prediction using p, and returns an estimate $\mathsf{LP}_{\ell,p}(\varphi(p, x))$ of the expected loss $\mathbb{E}[\ell(y, p(x))|x]$ suffered by p at the point x. We define a hierarchy of loss predictors of increasing strength, depending on the information contained in $\varphi(p, x)$:*

1. Prediction-only loss predictors *only have access to p's prediction at a point x, i.e. $\varphi(p, x) = p(x)$. The most natural choice for a prediction-only loss predictor is given by the self-entropy predictor, which we will define in Definition 3.*
2. Input-aware loss predictors *have access to the input features $i(x)$ used to train the model p, as well as its prediction, i.e. $\varphi(p, x) = (i(x), p(x))$.*
3. Representation-aware loss predictors *have access to $\varphi(p, x) = (p(x), i(x), r(x))$, where $r(x)$ is some representation of x. We distinguish between two kinds of representations:*
   - *Internal representations: The representation $r(x) = r_p(x)$ consists of features that are explicitly computed by the predictor p in the course of computing $p(x)$. For instance, they could consist of the embedding of x produced by the last few layers of a DNN.*
   - *External representations: The representation $r(x) = r_e(x)$ consists of features that are not explicitly computed by the predictor p. For instance, they could be the representation of x obtained from a different model, or by consulting human experts.*

[8] The case of general losses reduces to the proper loss case; please see Section A for details. We also assume for technical convenience that the loss is bounded. Losses that are not strictly bounded, such as cross entropy, can be handled with some further care and constraints on predicted probabilities.

A few comments on the definition:

- Two-headed architectures that simultaneously train both the predictor and the loss-predictor (such as [41, 28]) are a class of internal representation-aware predictors. In contrast, loss-predictors that use an embedding produced by a foundation model (such as [24], which audits the errors of the predictor) are external representation-aware.
- If we allow the loss predictor to be significantly more complex than the predictor p, then it could compute $r_p(x)$ from $i(x)$ using the model p. So the gap between input-aware and representation-aware loss predictors diminishes as the loss-predictor becomes more computationally powerful. But in the (important) setting where the loss predictor is less computationally powerful than the predictor, there could be a gap.
- In contrast, external representations might contain auxiliary information that cannot be computed using $i(x)$, regardless of the computational power of the loss predictor.

The loss predictor can be trained using standard regression, given access to a training set of points $(\varphi(p, x), y)$ where (x, y) are drawn from the distribution $\mathcal{D}$. One can measure the performance of our loss predictor as we would with any regression problem. We formulate it using the squared loss, as $\mathbb{E}[(\ell(y, p(x)) - \mathsf{LP}_{\ell,p}(\varphi(p, x))^2]$. It follows from Equation (1) that the Bayes optimal loss predictor is given by $\mathsf{LP}^*_{\ell,p}(x) = L(p^*(x); p(x))$. But computing this requires knowing the Bayes optimal predictor p^*, and is likely to be infeasible in most settings. Rather, we will compare our loss predictor to a canonical baseline which we describe next.

The self-entropy predictor

Following [7], given a predictor p, we define the simulated distribution $\mathcal{D}(p)$ on pairs $(x, \tilde{y}) \in \mathcal{X} \times \{0, 1\}$, where $x \sim \mathcal{D}$ and $\mathbb{E}[\tilde{y}|x] = p(x)$. The predictor hypothesizes that this how labels are being generated. Hence for each $x \in \mathcal{X}$, the self-entropy predictor predicts the expected loss according to this distribution.

▶ **Definition 3** (Self-entropy predictor)**.** *Given a proper loss ℓ and predictor p, the* self-entropy predictor *is the prediction-only loss predictor $\mathsf{SEP}_{\ell,p} : [0, 1] \to \mathbb{R}$ that predicts the expected loss when $\tilde{y} \sim \mathrm{Ber}(p(x))$ at each x; that is*

$$\mathsf{SEP}_{\ell,p}(p(x)) = \mathop{\mathbb{E}}_{\tilde{y}\sim\mathrm{Ber}(p(x))}[\ell(\tilde{y}, p(x))] = H_\ell(p(x)).$$

We use the self-entropy predictor as our baseline. Hence the question is when can we learn a loss predictor with significantly lower squared loss than the self-entropy predictor. We formalize this using the notion of advantage of a loss predictor over the self-entropy predictor.

▶ **Definition 4** (Advantage of a loss predictor)**.** *Define the advantage of a loss predictor $\mathsf{LP}_{\ell,p}$ over the self-entropy predictor to be the difference in the squared error*

$$\mathsf{adv}(\mathsf{LP}_{\ell,p}) = \mathbb{E}[(\ell(y, p(x)) - \mathsf{SEP}_{\ell,p}(p(x)))^2] - \mathbb{E}[(\ell(y, p(x)) - \mathsf{LP}_{\ell,p}(\varphi(p, x))^2].$$

We want loss predictors whose advantage is positive and as large as possible. Our goal is understand under what conditions we can hope to learn such a predictor.

On non-proper losses

So far we have assumed that we a trying to predict the proper loss incurred by a predictor. We can generalize this to a setting where we have a hypothesis $h : \mathcal{X} \to \mathcal{A}$ (for instance h might be a binary classifier), and a loss function $\ell : \{0,1\} \times \mathcal{A} \to \mathbb{R}$. It turns out that our theory extends seamlessly to the non-proper setting, under rather mild assumptions on the hypothesis h. We present this extension in Appendix A.

3 Multicalibration

Having defined our notion of a loss predictor, we next introduce the framework of multicalibration proposed by [20]. Our definition is most similar to the presentation used in [27].

▶ **Definition 5** (Multicalibration). *Let $\varphi(p,x) \in \Phi$ be some auxiliary set of features related to the computation of $p(x)$, which we define concretely below. Let $\mathcal{C}$ be a class of weight functions $c : \Phi \to [-1,1]$, and $p : \mathcal{X} \to [0,1]$ a binary predictor for a target distribution $\mathcal{D}$ over $\mathcal{X} \times \{0,1\}$. Then, the multicalibration error of p with respect to $\mathcal{C}$ is defined as*

$$\mathrm{MCE}(\mathcal{C}, p) := \max_{c \in \mathcal{C}} \left| \mathop{\mathbb{E}}_{x,y \sim \mathcal{D}} [(y - p(x)) c(\varphi(p,x))] \right|.$$

The information contained in $\varphi(p,x)$ gives rise to a hierarchy of multicalibration notions of increasing strength:

1. Calibration *corresponds to the setting where $\varphi(p,x) = p(x)$, and test functions can only depend on p's prediction.*
2. Multicalibration *corresponds to the case where test functions can additionally depend on the input features, i.e. $\varphi(p,x) = (p(x), i(x))$.*
3. Representation-aware multicalibration *is a strengthening of multicalibration where test functions can additionally depend on some representation $r(x)$ of x i.e., $\varphi(p,x) = (p(x), i(x), r(x))$. We distinguish between internal representations $r_p(x)$ and external representations $r_e(x)$ as with loss predictors (Definition 2).*

The first two levels in this hierarchy, calibration and multicalibration, have been extensively studied in previous works. In standard multicalibration, we require that a predictor $p(x)$ be well-calibrated under a broad class of test functions, $\mathcal{C}$, that depend only on $i(x)$ and $p(x)$. The literature on multicalibration typically identifies $i(x)$ with x itself. The last level of the hierarchy, representation-aware multicalibration, is a strengthening of multicalibration that naturally extends the multicalibration framework of [20]. As in the case of loss-predictors, the gap between internal representations $r_p(x)$ and $i(x)$ is computational; whereas the gap between external representations $r(x)$ and $i(x)$ could be information-theoretic.

▶ **Definition 6** (Multicalibration violation witness). *We say that a function $c : \Phi \times [0,1] \to [-1,1]$ is a witness for a multicalibration violation of magnitude α for a predictor p if*

$$\left| \mathop{\mathbb{E}}_{x,y \sim \mathcal{D}} [(y - p(x)) c(\varphi(p,x))] \right| > \alpha.$$

[20] showed that if we find such a witness, we can use it to improve the predictor p in a way that reduces the squared loss. While their argument is stated for the input-aware setting where $\varphi(p,x) = (p(x), i(x))$, it applies to the representation-aware setting as well.

4 Loss prediction advantage and multicalibration auditing

In this section, we establish the relationship between learning loss predictors with good advantage, and auditing for multicalibration, i.e. finding a c that witnesses a large multicalibration violation. The main result of our section is the following theorem, which establishes the correspondence between various levels of loss predictors and multicalibration requirements, when instantiated with the appropriate values $\varphi(p, x)$. We let $\Pi_{[0,1]} : \mathbb{R} \to [0, 1]$ denote the projection operator onto the unit interval.

▶ **Theorem 7.** *Let $\mathcal{F}$ be a class of loss predictors $f : \Phi \to [0, 1]$. Let $\mathcal{F}' \supseteq \mathcal{F}$ be the augmented function class defined as*

$$\mathcal{F}' = \{\Pi_{[0,1]}((1-\beta)H_\ell(p(x)) + \beta f(\varphi(p, x))) : \beta \in [-1, 1], f \in \mathcal{F}\}.$$

Let $\mathcal{C}$ be a class of weight functions defined as

$$\mathcal{C} = \{(f(\varphi(p, x)) - H_\ell(p(x)))H'_\ell(p(x)) : f \in \mathcal{F}\}.$$

Then,

$$\frac{1}{2} \max_{\mathsf{LP}_{\ell,p} \in \mathcal{F}} \mathsf{adv}(\mathsf{LP}_{\ell,p}) \leq \mathrm{MCE}(\mathcal{C}, p) \leq \sqrt{\max_{\mathsf{LP}_{\ell,p} \in \mathcal{F}'} \mathsf{adv}(\mathsf{LP}_{\ell,p})}.$$

The proof of Theorem 7 can be found in Appendix B.1 and follows from two key lemmas. Lemma 10 establishes the left-hand inequality by showing how a loss predictor with good advantage can be used to construct a witness of large multicalibration error. Conversely, Lemma 11 establishes the right-hand inequality by showing how to leverage a witness for large multicalibration error to construct a loss predictor with large advantage.

Before presenting our main lemmas, we introduce two auxiliary claims that are well-known in the literature on boosting and gradient descent. We provide proofs here for completeness and notational consistency.

Let $\mathcal{D}'$ be a distribution over $(x, z) \in X \times [0, 1]$. Let $h_1, h_2 : \mathcal{X} \to [0, 1]$ be two hypotheses. Under what conditions does h_2 improve on h_1? The following lemma gives a necessary condition: the update $\delta(x) = h_2(x) - h_1(x)$ must be correlated with the residual errors $z - h_1(x)$ of the hypothesis h_1 under the distribution $\mathcal{D}'$.

▷ Claim 8. For two hypotheses h_1, h_2,

$$\mathop{\mathbb{E}}_{\mathcal{D}'}[(h_1(x) - z)^2] - \mathop{\mathbb{E}}_{\mathcal{D}'}[(h_2(x) - z)^2] \leq 2 \mathop{\mathbb{E}}_{\mathcal{D}'}[(h_2(x) - h_1(x))(z - h_1(x))].$$

The proof of this claim can be found in Appendix B.2. Conversely, if we find an update $\delta(x)$ which is correlated with the residuals, we can perform a gradient descent update to reduce the squared error.

▷ Claim 9. If there exists $\delta : \mathcal{X} \to [-1, 1]$ such that $\mathbb{E}_{\mathcal{D}'}[\delta(x)(z - h_1(x))] \geq \beta \geq 0$, then setting $h_2(x) = \Pi_{[0,1]}(h_1(x) + \beta\delta(x))$ gives

$$\mathop{\mathbb{E}}_{\mathcal{D}'}[(h_1(x) - z)^2] - \mathop{\mathbb{E}}_{\mathcal{D}'}[(h_2(x) - z)^2] \geq \beta^2.$$

The proof of this claim can be found in Appendix B.3. With these claims in hand, we show that any loss predictor with a non-trivial advantage points us to a failure of multicalibration.

▶ **Lemma 10.** *Assume that* $\mathsf{LP}_{\ell,p}$ *achieves advantage* $\alpha \geq 0$ *over the self-entropy predictor. Then the function* $\delta(\varphi(p,x)) = \mathsf{LP}_{\ell,p}(\varphi(p,x)) - \mathsf{SEP}_{\ell,p}(p(x))$ *satisfies*

$$\mathbb{E}[\delta(\varphi(p,x))H'_\ell(p(x))(y-p(x))] \geq \alpha/2.$$

In other words, $\mathsf{LP}_{\ell,p}$ *can be used to construct a witness* $c(\varphi(p,x)) = \delta(\varphi(p,x))H'_\ell(p(x))$ *for a multicalibration violation of magnitude* $\alpha/2$*.*

The proof of this lemma can be found in Appendix B.4.

Conversely to the result of Lemma 10, we show that we can leverage certain types of multicalibration failures to predict loss with an advantage over the self-entropy predictor.

▶ **Lemma 11.** *Assume there exists a function* $\delta : \Phi \to [-1,1]$ *such that*

$$\mathbb{E}[\delta(\varphi(p,x))H'_\ell(p(x))(y-p(x))] \geq \beta \geq 0.$$

i.e., the function $c(\varphi(p,x)) = \delta(\varphi(p,x))H'_\ell(p(x))$ *is a witness for a multicalibration violation of magnitude* β*. Define the loss predictor*

$$\mathsf{LP}_{\ell,p}(\varphi(p,x)) = \Pi_{[0,1]}(\mathsf{SEP}_{\ell,p}(p(x)) + \beta\delta(\varphi(p,x))).$$

Then $\mathsf{adv}(\mathsf{LP}_{\ell,p}) \geq \beta^2$*.*

The proof of this lemma can be found in Appendix B.5.

5 Loss prediction for multiple losses

Up to this point, our discussion has focused on loss prediction for a single, predetermined loss function. However, in real-world applications, multiple stakeholders may use a predictor, each with unique objectives and priorities that correspond to different loss functions. This scenario would require training separate loss predictors for each user to meet their individual needs.

The self-entropy predictor offers a key advantage: it can be computed for any loss function using only the predictions $p(x)$, eliminating the need for additional training. Moreover, by extending the result of Theorem 7, we can define a class test functions $\mathcal{C}$ such that when p is multicalibrated with respect to $\mathcal{C}$, its self-entropy predictions simultaneously compete with the best-in-class loss predictor for each loss in a rich class of losses $\mathcal{L}$, rather than just a fixed loss. We formalize this in the following lemma, which we prove in Appendix C.1:

▶ **Lemma 12.** *Let* $\mathcal{F}$ *be a class of loss predictors* $f : \Phi \to [0,1]$*. Let* $\mathcal{L}$ *be a class of bounded proper losses* $\ell : \{0,1\} \times [0,1] \to [0,1]$ *with associated concave entropy functions* $H_\ell : [0,1] \to [0,1]$*, and let* $\mathcal{C}_\mathcal{L}$ *be the class of test functions*

$$\mathcal{C}_\mathcal{L} = \{(f(\varphi(p,x)) - H_\ell(p(x)))H'_\ell(p(x)) : f \in \mathcal{F}, \ell \in \mathcal{L}\}.$$

Then,

$$\max_{\ell\in\mathcal{L}} \max_{\mathsf{LP}_{\ell,p}\in\mathcal{F}} \mathsf{adv}(\mathsf{LP}_{\ell,p}) \leq 2\,\mathrm{MCE}(\mathcal{C}_\mathcal{L}, p).$$

I.e., no loss predictor from $\mathcal{F}$ *for any loss* $\ell \in \mathcal{L}$ *can obtain better advantage than* $2\,\mathrm{MCE}(\mathcal{C}_\mathcal{L}, p)$ *over the self-entropy predictor.*

When $\mathcal{L}$ is the set of all proper losses, the form of multicalibration imposed by $\mathcal{C}_{\mathcal{L}}$ can be thought of as the extension to multicalibration of the notion of *proper calibration*, recently proposed by [33]. The proper calibration error of a predictor p is defined as

$$\mathrm{PCE}(p) = \max_{\ell \in \mathcal{L}_{\mathrm{prop}}} |\mathbb{E}[H'_\ell(p(x))(y - p(x))]|$$

where $\mathcal{L}_{\mathrm{prop}}$ denotes the set of proper losses. Our condition can be thought of as "proper multicalibration" where each test function consists of $H'_\ell(p(x))$ multiplied with an additional test function $\delta(\varphi(p, x))$, that may depend on other features in addition to the prediction value.

5.1 Achieving efficient multicalibration for many losses

As the class of losses we consider expands, training an effective loss predictor for each individual loss becomes increasingly challenging. This section demonstrates that in certain scenarios, it is possible to efficiently produce a multicalibrated predictor with respect to the class of tests outlined in Lemma 12, even for some infinite classes of losses. This approach allows us to learn a single predictor p whose self-entropy estimates can compete with the best $\mathsf{LP}_{\ell,p} \in \mathcal{F}$ for every $\ell \in \mathcal{L}$, thus eliminating the need to train separate predictors for each loss.

This result draws on the techniques of [33], who show that when the class of functions $\{H'_\ell\}_{\ell \in \mathcal{L}}$ admits a finite approximate basis, proper calibration can be achieved efficiently. We present further discussion and a more general version of Theorem 14 in the full version of our paper (see "related version") and provide a simpler instantiation here for the class of 1-Lipschitz proper losses, $\mathcal{L}_{Lip}$.

We show efficiency in terms of oracle access to a weak-agnostic-learner for $\mathcal{F}$, the class of loss predictors. We motivate this assumption by observing that if we care about learning a loss predictor from the class $\mathcal{F}$, it's reasonable to assume that we have access to a weak agnostic learner for $\mathcal{F}$. We formally define a weak agnostic learner as follows:

▶ **Definition 13** (Weak agnostic learner). *Let $\alpha \geq 0$, $\delta \geq 0$. An α-weak agnostic learner for $\mathcal{F} \subseteq \{f : \Phi \to [-1, 1]\}$, closed under negation, with sample complexity n and failure parameter δ is an algorithm that when given n samples from a distribution $\mathcal{U}$ over $\Phi \times [-1, 1]$, outputs $f \in \mathcal{F} \cup \{\perp\}$ such that with probability at least $1 - \delta$ over the samples from $\mathcal{U}$ and the randomness in the algorithm itself, if $\max_{f \in \mathcal{F}} \mathbb{E}_{(\varphi,z)\sim\mathcal{U}}[f(\varphi)z] \geq \alpha$, the algorithm returns a $f \in \mathcal{F}$ such that $\mathbb{E}_{(\varphi,z)\sim\mathcal{U}}[f(\varphi)z] \geq \alpha/2$. Otherwise, if for all $f \in \mathcal{F}$, $\mathbb{E}_{(\varphi,z)\sim\mathcal{U}}[f(\varphi)z] \leq \alpha$, the algorithm either returns $f = \perp$ or $f \in \mathcal{F}$ such that $\mathbb{E}_{(\varphi,z)\sim\mathcal{U}}[f(\varphi)z] \geq \alpha/2$.*

With this definition in hand, we are ready to present the main theorem of this section. The proof can be found in the full version of the paper.

▶ **Theorem 14.** *Fix $\delta, \epsilon > 0$. Let $\mathcal{L}_{Lip}$ be the class of proper 1-Lipschitz losses $\ell : \{0, 1\} \times [0, 1] \to [0, 1]$, and let $\mathcal{F}$ be a class of loss predictors $\mathcal{F} : \Phi \to [-1, 1]$ that is closed under negation and contains the class of self entropy predictors, $\mathcal{H}_{\mathcal{L}_{Lip}} = \{H_\ell\}_{\ell \in \mathcal{L}_{Lip}}$. Further assume that we have access to an α-weak-agnostic-learner for $\mathcal{F}$ with sample complexity n and failure parameter $\beta \leq \frac{\alpha^2\delta}{4\lceil 2/\epsilon+1 \rceil}$.*

Then, there exists an algorithm that, given $m = O(n/\alpha^2)$ samples, with probability at least $1 - \delta$ outputs a predictor p such that

$$\max_{\ell \in \mathcal{L}_{Lip}} \max_{\mathsf{LP}_{\ell,p} \in \mathcal{F}} \mathsf{adv}(\mathsf{LP}_{\ell,p}) \leq 16\alpha + 4\epsilon.$$

In other words, our learned p's self-entropy predictions compete with the best-in-class loss predictor with *every* $\ell \in \mathcal{L}_{Lip}$, up to an error of $16\alpha + 4\epsilon$.

6 Experiments

We have shown in Section 4 a correspondence between the advantage a loss predictor has over the self entropy predictor and the multicalibration error. In this section, we empirically demonstrate this correspondence and see that it holds across several base models, loss prediction algorithms, as well as data subgroups.

Experiment design

We follow the basic design set forth in [19] for working with binary prediction tasks on UCI tabular datasets, specifically Credit Default [40] and Bank Marketing [32]. For each dataset, we consider certain subgroups (13 and 15 different groups respectively) defined by combinations of features such as occupation, education, and gender (see Appendix C.4 [19] for full details). These subgroups are used to evaluate the multicalibration error of the predictors as follows: for each subgroup we measure the *smoothed Expected Calibration Error* (smECE) [25], and take the multicalibration error to be the maximum smECE obtained.

We examine base predictors from different model families at various levels of multicalibration, specifically Naive Bayes and Support Vector Machines (SVMs), which tend to be uncalibrated without any postprocessing, along with Random Forests, Logistic Regression, Decision Trees, and Multilayer Perceptrons (MLPs), which tend to be well-calibrated out of the box when trained with empirical risk minimization. The base predictor MLP we use has a three-hidden-layer architecture with ReLU activations. For further details on hyperparameters, architectures, and training, we point the reader to Appendix E in [19].

We then run the following four loss prediction algorithms: decision tree regression, XGBoost, support vector regression (SVR), and a three-hidden-layer MLP. Each of these is input aware, that is, it is given both $i(x)$ and $p(x)$ at train time, and is trained using a standard regression objective to minimize $\mathbb{E}_{(x,y)\sim\mathcal{D}}[\big(\mathsf{LP}_{\ell,p}(i(x),p(x)) - \ell(y,p(x))\big)^2]$. Our target loss ℓ will be the squared loss $\ell(y,p(x)) = (y-p(x))^2$.

Results

Our main takeaways are as follows:

- Loss predictors perform better as the multicalibration error of the base model increases.
- Loss predictors perform better on subgroups that exhibit higher calibration error.

The first takeaway is demonstrated by Figure 1, where the horizontal axis indicates the max smECE of the base model, and the vertical axis indicates the advantage the loss predictor attains over the self-entropy predictor of the base model. As our theory predicts, we see a clear positive correlation between the maximum smECE and the relative performance of the loss predictor. In other words, less multicalibrated models have better performing loss predictors. This correlation holds across different base models and different algorithms for the loss predictor.

To delve deeper, we examine how loss prediction advantage varies across different subgroups. For this experiment we vary the base predictor only, while fixing the loss prediction algorithm to be an MLP. In Figure 2, for each base predictor and each subgroup, we report the loss predictor's advantage restricted to the subgroup on the vertical axis and the smECE of the subgroup on the horizontal axis.

For base models that are poorly calibrated overall (Decision Tree, SVM, and Naive Bayes), we see a clear correlation showing the loss predictor performs better on subgroups as the calibration error gets worse. By contrast, base predictors that are well-calibrated overall (Logistic Regression, Random Forest, and MLPs) allow negligible loss prediction advantage even after stratifying by subgroup.

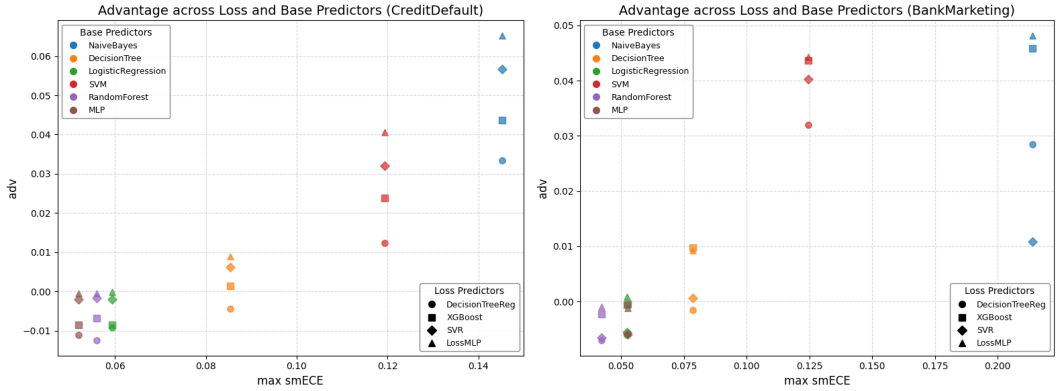

Figure 1 Advantage vs. max smECE across base predictors and loss prediction algorithms. For any particular loss predictor (shape), we see that as the multicalibration error of the base model (color) grows, the loss predictor's advantage improves.

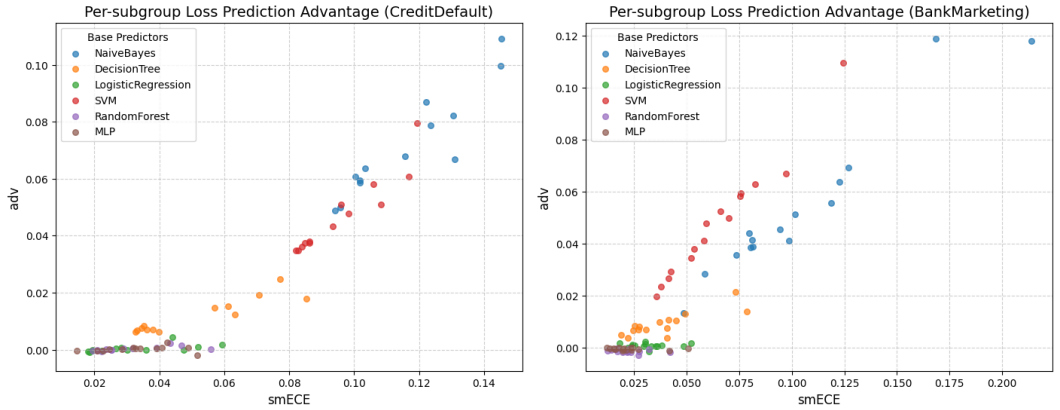

Figure 2 Fixing the type of loss predictor to be an MLP, we plot the loss advantage vs. the smECE on each subgroup across different base predictor models for the Credit Default and Bank Marketing datasets. For a fixed base predictor (color), the loss predictor exhibits more advantage on subgroups where the base predictor is less calibrated.

7 Related work

Applications of loss prediction

The idea of loss prediction has its roots in Bayesian and decision-theoretic active learning [38, 36], wherein the loss expected to be incurred at a point provides a natural measure of how valuable it is to label; see e.g. the well-known method of Expected Error Reduction (EER) [37]. To our knowledge, loss prediction in the explicit sense that we consider in this paper was first studied by [41], who proposed training an auxiliary loss prediction module alongside the base predictor. This is a practical approach used in many real-world applications. An important example from industry is the popular Segment Anything Model [28], which is an image segmentation model that includes an IoU prediction module[9]. This module plays a key role in the continual learning "data engine" used to train the model. Other applications of loss prediction are in routing inputs to weak or strong models [6, 34, 21], diagnosing model failures [24], and MRI reconstruction [22].

[9] IoU, or intersection over union, is a standard segmentation error metric.

Loss prediction is inherently connected to the broader topic of uncertainty quantification [23, 1]. The work of [31] formulates epistemic uncertainty as a form of excess loss or risk (see also [39]) and estimates it using an auxiliary loss predictor. Loss decomposition and connections to calibration have also been studied by [30, 2], although these works do not discuss auxiliary loss predictors per se.

Related theoretical work

We are not aware of prior theoretical work that studies the complexity of loss prediction. We build on notions and techniques from prior work on calibration [42, 18, 29], multicalibration [20, 27], omniprediction [16, 15, 17, 33] and outcome indistinguishability [7, 9]. Multicalibration has found applications to a myriad areas beyond multigroup fairness; a partial list includes omniprediction [16], domain adaptation [27], pseudorandomness [7, 10] and computational complexity [5]. Our work adds loss prediction to this list.

Decision calibration, decision OI and proper calibration

The work of [42] on decision calibration (implicitly) considered the self-entropy predictor, and conditions under which this predictor is accurate in expectation. The subsequent work of [15] termed this condition decision OI and showed that calibration of the predictor guarantees that the self-entropy predictor is itself calibrated for loss prediction. As we have seen, however, calibration of the predictor is not necessary for the self-entropy predictor to be calibrated (or optimal), due to the existence of blind-spots for a specific loss. This is explained by the notion of proper calibration introduced in [33], who showed that it tightly characterizes decision OI.

While our results have strong connections to all these works, the key difference is that our goal in loss prediction is not just to give loss estimates that are calibrated, it is to predict the true loss in the regression sense, which is potentially a much harder task.

Swap multicalibration

Our results equate the ability to gain an advantage over the self-predictor to a lack of multicalibration. This recalls a result of [17] which characterizes swap omniprediction by multicalibration: there too, the ability to achieve better loss than a simple baseline is attributable to a lack of multicalibration. Another related work is that of [12], which views multicalibration as a boosting algorithm for regression. Their work also connects multicalibration and regression, but the regression task they analyze is predicting y, whereas the regression task that we analyze is predicting $\ell(y, p(x))$.

Representation-aware multicalibration

Internal representation-aware multiaccuracy is considered in the work of [26] who use it in the context of face-recognition. Internal representation-aware multicalibration has connections to the notion of Code-Access Outcome Indistinguishability (OI), proposed by [8]. In Code-Access OI, outcomes generated by $p(x)$ must be indistinguishable from the true outcomes generated from the target distribution with respect to a set of tests that can inspect the full definition and code of p. With code access, such tests can compute the internal features $r_p(x)$ that are available in internal representation-aware multicalibration, but also have other capabilities such as querying p on other points $x' \in \mathcal{X}$. The use of external representations for auditing/improving predictions is found in the work of [4] who use skin type to assess

facial recognition; in recent work of [3], which investigates the use of expert opinions in addition to ML predictions to improve on medical test results, and in the work of [24] who use representations from a foundation model that is distinct from the model they audit.

Experimental work on multicalibration

The work of [26] considers internal-representation aware multiaccuracy for facial recognition tasks, and shows how auditing can be used to improve accuracy across subgroups. Recently, [19] conducted a systematic investigation of multicalibration in practice, analyzing the utility of algorithms for multicalibration on a number of real-world datasets, prediction models, and demographic subgroups. We build closely on their setup for our own experiments.

References

1 Moloud Abdar, Farhad Pourpanah, Sadiq Hussain, Dana Rezazadegan, Li Liu, Mohammad Ghavamzadeh, Paul Fieguth, Xiaochun Cao, Abbas Khosravi, U Rajendra Acharya, et al. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information fusion*, 76:243–297, 2021. doi:10.1016/J.INFFUS.2021.05.008.

2 Gustaf Ahdritz, Aravind Gollakota, Parikshit Gopalan, Charlotte Peale, and Udi Wieder. Provable uncertainty decomposition via higher-order calibration. In *International Conference on Learning Representations*, 2025. To appear.

3 Rohan Alur, Loren Laine, Darrick K. Li, Dennis Shung, Manish Raghavan, and Devavrat Shah. Integrating expert judgment and algorithmic decision making: An indistinguishability framework, 2024. doi:10.48550/arXiv.2410.08783.

4 Joy Buolamwini and Timnit Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on Fairness, Accountability and Transparency, FAT 2018, 23-24 February 2018, New York, NY, USA*, volume 81 of *Proceedings of Machine Learning Research*, pages 77–91. PMLR, 2018. URL: http://proceedings.mlr.press/v81/buolamwini18a.html.

5 Sílvia Casacuberta, Cynthia Dwork, and Salil P. Vadhan. Complexity-theoretic implications of multicalibration. In *56th Annual ACM Symposium on Theory of Computing, STOC 2024*, pages 1071–1082. ACM, 2024. doi:10.1145/3618260.3649748.

6 Dujian Ding, Ankur Mallick, Chi Wang, Robert Sim, Subhabrata Mukherjee, Victor Rühle, Laks VS Lakshmanan, and Ahmed Hassan Awadallah. Hybrid llm: Cost-efficient and quality-aware query routing. In *The Twelfth International Conference on Learning Representations*, 2024.

7 Cynthia Dwork, Michael P. Kim, Omer Reingold, Guy N. Rothblum, and Gal Yona. Outcome indistinguishability. In *ACM Symposium on Theory of Computing (STOC'21)*, 2021. arXiv:2011.13426.

8 Cynthia Dwork, Michael P Kim, Omer Reingold, Guy N Rothblum, and Gal Yona. Outcome indistinguishability. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 1095–1108, 2021. doi:10.1145/3406325.3451064.

9 Cynthia Dwork, Michael P. Kim, Omer Reingold, Guy N. Rothblum, and Gal Yona. Beyond bernoulli: Generating random outcomes that cannot be distinguished from nature. In *The 33rd International Conference on Algorithmic Learning Theory*, 2022.

10 Cynthia Dwork, Daniel Lee, Huijia Lin, and Pranay Tankala. From pseudorandomness to multi-group fairness and back. In *36th Annual Conference on Learning Theory, COLT 2023*, volume 195 of *Proceedings of Machine Learning Research*, pages 3566–3614. PMLR, 2023. URL: https://proceedings.mlr.press/v195/dwork23a.html.

11 Dean Foster and Rakesh Vohra. Regret in the on-line decision problem. *Games and Economic Behavior*, 29:7–35, 1999.

12 Ira Globus-Harris, Declan Harrison, Michael Kearns, Aaron Roth, and Jessica Sorrell. Multicalibration as boosting for regression. In *International Conference on Machine Learning, ICML 2023*, 2023.

13 Tilmann Gneiting and Adrian E Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American statistical Association*, 102(477):359–378, 2007.

14 Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS'14*, pages 2672–2680, 2014. URL: `https://proceedings.neurips.cc/paper/2014/hash/5ca3e9b122f61f8f06494c97b1afccf3-Abstract.html`.

15 Parikshit Gopalan, Lunjia Hu, Michael P. Kim, Omer Reingold, and Udi Wieder. Loss minimization through the lens of outcome indistinguishability. In *Innovations in theoretical computer science (ITCS'23)*, 2023.

16 Parikshit Gopalan, Adam Tauman Kalai, Omer Reingold, Vatsal Sharan, and Udi Wieder. Omnipredictors. In *Innovations in Theoretical Computer Science (ITCS'2022)*, 2022. `arXiv:2109.05389`.

17 Parikshit Gopalan, Michael P. Kim, and Omer Reingold. Swap agnostic learning, or characterizing omniprediction via multicalibration. In *NeurIPS'23*, 2023.

18 Parikshit Gopalan, Michael P Kim, Mihir A Singhal, and Shengjia Zhao. Low-degree multicalibration. In *Conference on Learning Theory*, pages 3193–3234. PMLR, 2022. URL: `https://proceedings.mlr.press/v178/gopalan22a.html`.

19 Dutch Hansen, Siddartha Devic, Preetum Nakkiran, and Vatsal Sharan. When is Multicalibration Post-Processing Necessary? In *Advances in Neural Information Processing Systems*, 2024.

20 Ursula Hébert-Johnson, Michael Kim, Omer Reingold, and Guy Rothblum. Multicalibration: Calibration for the (computationally-identifiable) masses. In *International Conference on Machine Learning*, pages 1939–1948. PMLR, 2018.

21 Qitian Jason Hu, Jacob Bieker, Xiuyu Li, Nan Jiang, Benjamin Keigwin, Gaurav Ranganath, Kurt Keutzer, and Shriyash Kaustubh Upadhyay. Routerbench: A benchmark for multi-llm routing system. *arXiv preprint arXiv:2403.12031*, 2024. `doi:10.48550/arXiv.2403.12031`.

22 Shi Hu, Nicola Pezzotti, and Max Welling. Learning to predict error for mri reconstruction. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2021*, pages 604–613. Springer, 2021. `doi:10.1007/978-3-030-87199-4_57`.

23 Eyke Hüllermeier and Willem Waegeman. Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine learning*, 110(3):457–506, 2021. `doi:10.1007/S10994-021-05946-3`.

24 Saachi Jain, Hannah Lawrence, Ankur Moitra, and Aleksander Madry. Distilling model failures as directions in latent space. In *International Conference on Learning Representations*, 2023.

25 Błasiok Jarosław and Preetum Nakkiran. Smooth ECE: Principled Reliability Diagrams via Kernel Smoothing. In *International Conference on Learning Representations*, 2024.

26 Michael P. Kim, Amirata Ghorbani, and James Zou. 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, 2019. `doi:10.1145/3306618.3314287`.

27 Michael P Kim, Christoph Kern, Shafi Goldwasser, Frauke Kreuter, and Omer Reingold. Universal adaptability: Target-independent inference that competes with propensity scoring. *Proceedings of the National Academy of Sciences*, 119(4), 2022.

28 Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023.

29 Bobby Kleinberg, Renato Paes Leme, Jon Schneider, and Yifeng Teng. U-calibration: Forecasting for an unknown agent. In *The Thirty Sixth Annual Conference on Learning Theory*, pages 5143–5145. PMLR, 2023. URL: `https://proceedings.mlr.press/v195/kleinberg23a.html`.

30 Meelis Kull and Peter Flach. Novel decompositions of proper scoring rules for classification: Score adjustment as precursor to calibration. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2015*, pages 68–85. Springer, 2015. doi:10.1007/978-3-319-23528-8_5.

31 Salem Lahlou, Moksh Jain, Hadi Nekoei, Victor Ion Butoi, Paul Bertin, Jarrid Rector-Brooks, Maksym Korablyov, and Yoshua Bengio. DEUP: Direct epistemic uncertainty prediction. *arXiv preprint arXiv:2102.08501*, 2021.

32 S. Moro, P. Rita, and P. Cortez. Bank Marketing. UCI Machine Learning Repository, 2014. DOI: https://doi.org/10.24432/C5K306.

33 Princewill Okoroafor, Robert Kleinberg, and Michael P Kim. Near-optimal algorithms for omniprediction. *arXiv preprint arXiv:2501.17205*, 2025. doi:10.48550/arXiv.2501.17205.

34 Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E Gonzalez, M Waleed Kadous, and Ion Stoica. Routellm: Learning to route llms with preference data. *arXiv preprint arXiv:2406.18665*, 2024. doi:10.48550/arXiv.2406.18665.

35 Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

36 Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Brij B Gupta, Xiaojiang Chen, and Xin Wang. A survey of deep active learning. *ACM computing surveys (CSUR)*, 54(9):1–40, 2021. doi:10.1145/3472291.

37 Nicholas Roy and Andrew McCallum. Toward optimal active learning through monte carlo estimation of error reduction. In *International Conference on Machine Learning*, 2001.

38 Burr Settles. Active learning literature survey, 2009.

39 Aolin Xu and Maxim Raginsky. Minimum excess risk in bayesian learning. *IEEE Transactions on Information Theory*, 68(12):7935–7955, 2022. doi:10.1109/TIT.2022.3176056.

40 I-Cheng Yeh. Default of Credit Card Clients. UCI Machine Learning Repository, 2009. DOI: https://doi.org/10.24432/C55S3H.

41 Donggeun Yoo and In So Kweon. Learning loss for active learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 93–102, 2019. doi:10.1109/CVPR.2019.00018.

42 Shengjia Zhao, Michael P. Kim, Roshni Sahoo, Tengyu Ma, and Stefano Ermon. Calibrating predictions to decisions: A novel approach to multi-class calibration. In *Advances in Neural Information Processing Systems*, 2021. URL: https://openreview.net/forum?id=iFF-zKCgzS.

A Handling non-proper losses

We consider an abstract action space $\mathcal{A}$; examples are the discrete setting where $\mathcal{A} = [k]$, and the continuous setting where $\mathcal{A} = \mathbb{R}$. A hypothesis is a function $h : \mathcal{X} \to \mathcal{A}$. A loss function is a function $\ell : \{0,1\} \times \mathcal{A} \to [0,1]$. The expected loss of hypothesis h at the point x is given by $\mathbb{E}[\ell(y, h(x))|x]$. The goal of a loss predictor is to learn a function $\mathsf{LP}_{\ell,p} : \Phi \to \mathbb{R}$ that gives pointwise estimates of this quantity. As in definition 2, we can define a hierarchy of loss predictors based on the features available to them.

For any loss ℓ, if the labels are drawn according to $y \sim \mathrm{Ber}(p)$ for any $p \in [0,1]$, then the optimal prediction that minimizes the loss, $k_\ell(p) \in [0,1]$ is defined by

$$k_\ell(p) = \arg\min_{v \in [0,1]} \mathop{\mathbb{E}}_{y \sim \mathrm{Ber}(p^*(x))} [\ell(y, v)]^{10}.$$

If there exist a *latent* predictor $p_h : \mathcal{X} \to [0,1]$ so that $h = k_\ell \circ p_h$ is obtained by best-responding to its predictions, then we can reduce to the setting of proper losses, since

$$\mathbb{E}[\ell(y, h(x))] = \mathbb{E}[\ell(y, k_\ell(p(x)) = \mathbb{E}[\ell \circ k_\ell(y, p(x))]$$

and we have the following result of [29].

▶ **Lemma 15** ([29]). *For any loss $\ell : \{0,1\} \times \mathcal{A} \to [0,1]$, the loss function $\ell \circ k_\ell : \{0,1\} \times [0,1] \to \mathbb{R}$ is a proper loss.*

But under what conditions on h does there exist such a predictor p_h? And is it easy to estimate its predictions given access to h?

To answer the first question, we show that it is equivalent to assuming that the hypothesis satisfies a simple optimality condition for the loss.

▶ **Definition 16.** *The hypothesis $h : \mathcal{X} \to [0,1]$ is swap-optimal for $\mathcal{D}$ if for every function $\kappa : \mathcal{A} \to \mathcal{A}$, it holds that $\mathbb{E}[\ell(y, h(x))] \leq \mathbb{E}[\ell(y, \kappa(h(x)))]$.*

Swap optimality is a weak optimality condition that can be easily achieved by post-processing. It is quite reasonable to assume that a well-trained model optimized to minimize loss satisfies this guarantee. For instance, a well-trained image classifier should not improve if every time it predicts *cat*, we say *dog* instead. For a swap optimal hypothesis h, we show that is indeed easy to identify a latent predictor p_h so that h is obtained by best-responding to its predictions. This theorem lets us extend our theory of loss prediction for proper losses to arbitrary loss functions under the rather weak assumption that h is swap-optimal.

▶ **Theorem 17.** *Given a hypothesis $h : \mathcal{X} \to \mathcal{A}$ and a distribution $\mathcal{D}$, define the predictor $p_h : \mathcal{X} \to [0,1]$ by $p_h(x) = \mathbb{E}_{\mathcal{D}}[y|h(x)]$. The hypothesis h is swap optimal iff $h(x) = k_\ell \circ p_h(x)$ for all $x \in \mathcal{X}$.*[11]

Proof. Assume that h is not swap-optimal, so there exist κ such that $\mathbb{E}[\ell(y, \kappa(h(x)))] < \mathbb{E}[\ell(y, h(x))]$. There must exist a specific choice of $h(x) = a \in \mathcal{A}$ conditioned on which the inequality still holds, hence

$$\mathbb{E}[\ell(y, \kappa(a))|h(x) = a] \leq \mathbb{E}[\ell(y, a)|h(x) = a].$$

Let $\mathbb{E}[y|h(x) = a] = v$. But this shows that when $y \sim \mathrm{Ber}(v)$, $\mathbb{E}[\ell(y, \kappa(a)] < \mathbb{E}[\ell(y, a)]$, so $a \neq k_\ell(v)$. Hence for all $x \in h^{-1}(a)$, we have $h(x) = a \neq k_\ell(v) = k_\ell(p_h(x))$.

Conversely, if h is indeed swap optimal, then it must be the case that every action $a \in \mathcal{A}$ is a best response to $\mathbb{E}[y|h(x) = a] = p_h(x)$, which means we have $h(x) = k_\ell(p_h(x))$. ◀

B Proofs from Section 4

B.1 Proof of Theorem 7

Proof of Theorem 7. The inequality on the left follows from Theorem 10, while the inequality on the right follows from Lemma 11. We prove each in turn, starting with the left-hand inequality.

By Theorem 10, if there exists a $f \in \mathcal{F}$ such that setting $\mathsf{LP}_{\ell,p} = f$ gives $\mathsf{adv}(\mathsf{LP}_{\ell,p}) = \alpha$, then this implies that

$$\mathbb{E}[(\mathsf{LP}_{\ell,p}(\varphi(p,x) - \mathsf{SEP}_{\ell,p}(p(x)))H'_\ell(p(x))(y - p(x))] \geq \alpha/2.$$

[11] Strictly speaking, k_ℓ is not a function as there can be many optimal actions. However we interpret this equation as saying $h(x)$ is in the set of optimal actions for $p_h(x)$.

We observe that because $\mathsf{LP}_{\ell,p} = f \in \mathcal{F}$, the witness of this multicalibration violation, $(\mathsf{LP}_{\ell,p}(\varphi(p,x) - \mathsf{SEP}_{\ell,p}(p(x)))H'_\ell(p(x))$ lies in $\mathcal{C}$, and thus

$$\begin{aligned}\mathrm{MCE}(\mathcal{C},p) &= \max_{c\in\mathcal{C}} |\mathbb{E}[c(\varphi(p,x))(y-p(x))]| \\ &\geq \mathbb{E}[(\mathsf{LP}_{\ell,p}(\varphi(p,x) - \mathsf{SEP}_{\ell,p}(p(x)))H'_\ell(p(x))(y-p(x))] \\ &\geq \alpha/2 \\ &= \frac{1}{2}\mathsf{adv}(\mathsf{LP}_{\ell,p})\end{aligned}$$

The inequality follows by taking the maximum over all $\mathsf{LP}_{\ell,p} \in \mathcal{F}$, as $\mathsf{LP}_{\ell,p}$ was chosen arbitrarily.

We now move on to proving the inequality on the right, i.e., the upper bound on $\mathrm{MCE}(\mathcal{C},p)$.

By definition of the multicalibration error and $\mathcal{C}$, there exists some $c \in \mathcal{C}$ that witnesses a multicalibration error of magnitude $\mathrm{MCE}(\mathcal{C},p)$, i.e. for some $f \in \mathcal{F}$,

$$\mathrm{MCE}(\mathcal{C},p) = \left|\underbrace{\mathbb{E}[(f(\varphi(p,x)) - H_\ell(p(x)))H'_\ell(p(x))(y-p(x))]}_{:=E_f}\right|.$$

Thus, if we define $\delta : \Phi \to [-1,1]$ as

$$\delta(\varphi(p,x)) = \mathrm{sign}(E_f)(f(\varphi(p,x)) - H_\ell(p(x))),$$

it follows that

$$\mathbb{E}[\delta(\varphi(p,x))H'_\ell(p(x))(y-p(x))] = \mathrm{MCE}(\mathcal{C},p).$$

Applying Lemma 11 for this δ implies that for the loss predictor defined by $\mathsf{LP}_{\ell,p}(\varphi(p,x)) = \Pi_{[0,1]}(\mathsf{SEP}_{\ell,p}(p(x)) + \mathrm{MCE}(\mathcal{C},p)\delta(\varphi(p,x)))$ satisfies

$$\mathsf{adv}(\mathsf{LP}_{\ell,p}(\varphi(p,x))) \geq \mathrm{MCE}(\mathcal{C},p)^2.$$

The proof of the inequality follows by observing that $\mathsf{LP}_{\ell,p} \in \mathcal{F}'$, because

$$\begin{aligned}\mathsf{LP}_{\ell,p}(\varphi(p,x)) &= \Pi_{[0,1]}(\mathsf{SEP}_{\ell,p}(p(x)) + \mathrm{MCE}(\mathcal{C},p)\delta(\varphi(p,x))) \\ &= \Pi_{[0,1]}(H_\ell(p(x)) + \underbrace{\mathrm{MCE}(\mathcal{C},p)\,\mathrm{sign}(E_f)}_{:=\beta}(f(\varphi(p,x)) - H_\ell(p(x)))) \\ &= \Pi_{[0,1]}((1-\beta)H_\ell(p(x)) + \beta f(\varphi(p,x)))\end{aligned}$$

Where $\beta = \mathrm{sign}(E_f)\,\mathrm{MCE}(\mathcal{C},p) \in [-1,1]$, because $\mathrm{MCE}(\mathcal{C},p) \in [0,1]$.

Thus, $\mathsf{LP}_{\ell,p} \in \mathcal{F}'$, and so we conclude that

$$\max_{\mathsf{LP}_{\ell,p}\in\mathcal{F}'} \mathsf{adv}(\mathsf{LP}_{\ell,p}(\varphi(p,x))) \geq \mathrm{MCE}(\mathcal{C},p)^2.$$

We get the right-hand inequality from the statement after taking square root of both sides. ◀

B.2 Proof of Claim 8

Proof of Claim 8. Let us write $\delta(x) = h_2(x) - h_1(x)$. Then we have

$$\begin{aligned}\mathop{\mathbb{E}}_{\mathcal{D}'}[(h_1(x)-z)^2] - \mathop{\mathbb{E}}_{\mathcal{D}'}[(h_2(x)-z)^2] &= \mathop{\mathbb{E}}_{\mathcal{D}'}[(h_1(x)-z)^2 - (h_1(x)-z+\delta(x))^2] \\ &= -2\mathop{\mathbb{E}}_{\mathcal{D}'}[(h_1(x)-z)\delta(x)] - \mathop{\mathbb{E}}_{\mathcal{D}'}[\delta(x)^2] \\ &\leq 2\mathop{\mathbb{E}}_{\mathcal{D}'}[(z-h_1(x))\delta(x)].\end{aligned}$$

◁

B.3 Proof of Claim 9

Proof of Claim 9. Without projection, we can write the gap in squared error as

$$\begin{aligned}\mathop{\mathbb{E}}_{\mathcal{D}'}[(h_1(x)-z)^2] - \mathop{\mathbb{E}}_{\mathcal{D}'}[(h_1(x)+\beta\delta(x)-z)^2] &= 2\beta\mathop{\mathbb{E}}_{\mathcal{D}'}[(z-h_1(x))\delta(x)] - \beta^2\mathop{\mathbb{E}}_{\mathcal{D}'}[\delta(x)^2] \\ &\geq 2\beta^2 - \beta^2 = \beta^2.\end{aligned}$$

While $h_1(x) + \beta\delta(x)$ may not be bounded in $[0,1]$, projection onto the interval can only further reduce the squared error. ◁

B.4 Proof of Lemma 10

Proof of Lemma 10. Consider the loss regression problem, where we draw $(x, y) \in \mathcal{X} \times \{0,1\} \sim \mathcal{D}$ and then return the pair $(x, z = \ell(y, p(x))$. We will use Claim 8, where we take $h_1 = \mathsf{SEP}_{\ell,p}$ to be the self-entropy predictor and $h_2 = \mathsf{LP}_{\ell,p}$. We can estimate the residual error of the self-entropy predictor as

$$\begin{aligned}\ell(y,p(x)) - \mathsf{SEP}_{\ell,p}(p(x)) &= H_\ell(p(x)) + (y-p(x))H'_\ell(p(x)) - H_\ell(p(x)) \\ &= (y-p(x))H'_\ell(p(x)).\end{aligned} \tag{2}$$

By Claim 8, we have

$$\begin{aligned}\alpha &= \mathbb{E}[(\ell(y,p(x)) - \mathsf{SEP}_{\ell,p}(p(x)))^2] - \mathbb{E}[(\ell(y,p(x)) - \mathsf{LP}_{\ell,p}(\varphi(p,x))^2] \\ &\leq 2\,\mathbb{E}[(\mathsf{LP}_{\ell,p}(\varphi(p,x)) - \mathsf{SEP}_{\ell,p}(p(x)))(\ell(y,p(x)) - \mathsf{SEP}_{\ell,p}(p(x)))] \\ &= 2\,\mathbb{E}[\delta(\varphi(p,x))H'_\ell(p(x))(y-p(x))]\end{aligned}$$

◀

B.5 Proof of Lemma 11

Proof of Lemma 11. We again consider the loss regression problem, We now apply Lemma 9 with $z = \ell(y, p(x))$, $h_1 = \mathsf{SEP}_{\ell,p}$. The correlation condition we require is

$$\mathbb{E}[\delta(\varphi(p,x))(\ell(y,p(x) - \mathsf{SEP}_{\ell,p}(p(x)))] \geq \beta.$$

By Equation (2), we have

$$\mathbb{E}[\delta(\varphi(p,x))(\ell(y,p(x)) - \mathsf{SEP}_{\ell,p}(p(x)))] = \mathbb{E}[\delta(\varphi(p,x))H'_\ell(p(x))(y-p(x))]$$

which is at least β by our assumption. Hence Claim 9 implies that $h_2 = \mathsf{LP}_{\ell,p}$ has advantage β^2 over $\mathsf{SEP}_{\ell,p}$. ◀

C Proofs from Section 5

C.1 Proof of Lemma 12

Proof of Lemma 12. For a fixed loss $\ell \in \mathcal{L}$, let

$$\mathcal{C}_\ell = \{(f(\varphi(p,x)) - H_\ell(p(x)))H'_\ell(p(x)) : f \in \mathcal{F}\}.$$

By Theorem 7, we can guarantee that

$$\max_{\mathsf{LP}_{\ell,p} \in \mathcal{F}} \mathsf{adv}(\mathsf{LP}_{\ell,p}) \leq 2\,\mathrm{MCE}(\mathcal{C}_\ell, p).$$

Taking the max over $\mathcal{L}$ for both sides, we get

$$\max_{\ell \in \mathcal{L}} \max_{\mathsf{LP}_{\ell,p} \in \mathcal{F}} \mathsf{adv}(\mathsf{LP}_{\ell,p}) \leq \max_{\ell \in \mathcal{L}} 2\,\mathrm{MCE}(\mathcal{C}_\ell, p) \leq 2\,\mathrm{MCE}(\mathcal{C}_\mathcal{L}, p).$$

Where the right-most inequality follows from the fact that $\mathcal{C}_\mathcal{L} = \bigcup_{\ell \in \mathcal{L}} \mathcal{C}_\ell$. This proves the desired inequality. ◀