# Calibrating Predictions to Decisions: A Novel Approach to Multi-Class Calibration

**Shengjia Zhao**
Stanford University
sjzhao@stanford.edu

**Michael P. Kim**
UC Berkeley
mpkim@berkeley.edu

**Roshni Sahoo**
Stanford University
rsahoo@stanford.edu

**Tengyu Ma**
Stanford University
tengyuma@stanford.edu

**Stefano Ermon**
Stanford University
ermon@stanford.edu

## Abstract

When facing uncertainty, decision-makers want predictions they can trust. A machine learning provider can convey confidence to decision-makers by guaranteeing their predictions are distribution calibrated — amongst the inputs that receive a predicted class probabilities vector $q$, the actual distribution over classes is $q$. For multi-class prediction problems, however, achieving distribution calibration tends to be infeasible, requiring sample complexity exponential in the number of classes $C$. In this work, we introduce a new notion—*decision calibration*—that requires the predicted distribution and true distribution to be "indistinguishable" to a set of downstream decision-makers. When all possible decision makers are under consideration, decision calibration is the same as distribution calibration. However, when we only consider decision makers choosing between a bounded number of actions (e.g. polynomial in $C$), our main result shows that decisions calibration becomes feasible — we design a recalibration algorithm that requires sample complexity polynomial in the number of actions and the number of classes. We validate our recalibration algorithm empirically: compared to existing methods, decision calibration improves decision-making on skin lesion and ImageNet classification with modern neural network predictors.

## 1 Introduction

Machine learning predictions are increasingly employed by downstream decision makers who have little or no visibility on how the models were designed and trained. In high-stakes settings, such as healthcare applications, decision makers want predictions they can trust. For example in healthcare, suppose a machine learning service offers a supervised learning model to healthcare providers that claims to predict the probability of various skin diseases, given an image of a lesion. Each healthcare provider want assurance that the model's predictions lead to beneficial decisions, according to their own loss functions. As a result, the healthcare providers may reasonably worry that the model was trained using a loss function different than their own. This mismatch is often inevitable because the ML service may provide the same prediction model to many healthcare providers, which may have different treatment options available and loss functions. Even the same healthcare provider could have different loss functions throughout time, due to changes in treatment availability.

If predicted probabilities perfectly equal the true probability of the event, this issue of trust would not arise because they would lead to optimal decision making regardless of the loss function or task considered by downstream decision makers. In practice, however, predicted probabilities are never perfect. To address this, the healthcare providers may insist that the prediction function be

35th Conference on Neural Information Processing Systems (NeurIPS 2021).

*distribution calibrated*, requiring that amongst the inputs that receive predicted class probability vectors $q$, the actual distribution over classes is $q$. This solves the trust issue because among the patients who receive prediction $q$, the healthcare providers knows that the true label distribution is $q$, and hence knows the true expected loss of a treatment on these patients. Unfortunately, to achieve distribution calibration, we need to reason about the set of individuals $x$ who receive prediction $q$, for *every* possible predicted $q$. As the number of distinct predictions may naturally grow exponentially in the number of classes $C$, the amount of data needed to accurately certify distribution calibration tends to be prohibitive. Due to this statistical barrier, most work on calibrated multi-class predictions focuses on obtaining relaxed variants of calibration. These include *confidence calibration* (Guo et al., 2017), which calibrates predictions only over the most likely class, and *classwise calibration* (Kull et al., 2019), which calibrates predictions for each class marginally. While feasible, these notions are significantly weaker than distribution calibration and do not address the trust issue highlighted above. Is there a calibration notion that addresses the issue of trust, but can also be verified and achieved efficiently? Our paper answers this question affirmatively.

**Our Contributions.** We introduce a new notion of calibration—*decision calibration*—where we take the perspective of potential decision-makers: the only differences in predictions that matter are those that could lead to different decisions. Inspired by Dwork et al. (2021), we formalize this intuition by requiring that predictions are "indistinguishable" from the true outcomes, according to a collection of decision-makers.

First, we show that prior notions of calibration can be characterized as special cases of decision calibration under different collections of decision-makers. This framing explains the strengths and weakness of existing notions of calibration, and clarifies the guarantees they offer to decision makers. For example, we show that a predictor is distribution calibrated if and only if it is decision calibrated with respect to *all* loss functions and decision rules. This characterization demonstrates why distribution calibration is so challenging: achieving distribution calibration requires simultaneously reasoning about all possible decision tasks.

The set of *all* decision rules include those that choose between exponentially (in number of classes $C$) many actions. In practice, however, decision-makers typically choose from a bounded (or slowly-growing as a function of $C$) set of actions. Our main contribution is an algorithm that *guarantees decision calibration for such more realistic decision-makers*. In particular, we give a sample-efficient algorithm that takes a pre-trained predictor and post-processes it to achieve decision calibration with respect to *all* decision-makers choosing from a bounded set of actions. Our recalibration procedure does not harm other common performance metrics, and actually improves accuracy and likelihood of the predictions. In fact, we argue formally that, in the setting of bounded actions, optimizing for decision calibration recovers many of the benefits of distribution calibration, while drastically improving the sample complexity. Empirically, we use our algorithm to recalibrate deep network predictors on two large scale datasets: skin lesion classification (HAM10000) and Imagenet. Our recalibration algorithm improves decision making, and allow for more accurate decision loss estimation compared to existing recalibration methods even under distribution shift.

## 2 Background

### 2.1 Setup and Notation

We consider the prediction problem with random variables $X$ and $Y$, where $X \in \mathcal{X}$ denotes the input features, and $Y \in \mathcal{Y}$ denotes the label. We focus on classification where $\mathcal{Y} = \{(1, 0, \cdots, 0), (0, 1, \cdots, 0), \cdots, (0, 0, \cdots, 1)\}$ where each $y \in \mathcal{Y}$ is a one-hot vector with $C \in \mathbb{N}$ classes. [1] A probability prediction function is a map $\hat{p} : \mathcal{X} \to \Delta^C$ where $\Delta^C$ is the $C$-dimensional simplex. We define the support of $\hat{p}$ as the set of distributions it could predict, i.e. $\{\hat{p}(x) | x \in \mathcal{X}\}$. We use $p^* : \mathcal{X} \to \Delta^C$ to denote the true conditional probability vector, i.e. $p^*(x) = \mathbb{E}[Y \mid X = x] \in \Delta^C$, and for all $c \in [C]$, each coordinate gives the probability of the class $p^*(x)_c = \Pr[Y_c \mid X = x]$.

[1] We can also equivalently define $\mathcal{Y} = \{1, 2, \cdots, C\}$, here we denote $y$ by a one-hot vector for notation convenience when taking expectations.

### 2.2 Decision-Making Tasks and Loss Functions

We formalize a decision-making task as a loss minimization problem. The decision-maker has some set of available actions $\mathcal{A}$ and a loss function $\ell : \mathcal{Y} \times \mathcal{A} \to \mathbb{R}$. In this paper we assume the loss function does not directly depend on the input features $X$. For notational simplicity we often refer to a (action set, loss function) pair $(\mathcal{A}, \ell)$ only by the loss function $\ell$: the set of actions $\mathcal{A}$ is implicitly defined by the domain of $\ell$. We denote the set of all possible loss functions as $\mathcal{L}_{\text{all}} = \{\ell | \forall \text{ set } \mathcal{A} \text{ and function } \ell : \mathcal{Y} \times \mathcal{A} \to \mathbb{R}\}$.

We treat all action sets $\mathcal{A}$ with the same cardinality as the same set — they are equivalent up to renaming the actions. A convenient way to think about this is that we only consider actions sets $\mathcal{A} \in \{[1], [2], \cdots, [K], \cdots, \mathbb{N}, \mathbb{R}\}$ where $[K] = \{1, \cdots, K\}$.

### 2.3 Bayes Decision-Making

Given some predicted probability $\hat{p}(X)$ on $Y$, a decision-maker selects an action in $\mathcal{A}$. We assume that the decision-maker selects the action based on the predicted probability. That is, we define a decision function as any map from the predicted probability to an action $\delta : \Delta^C \to \mathcal{A}$. and denote by $\mathbb{\Delta}_{\text{all}}$ as the set of all decision functions with any set of actions $\mathbb{\Delta}_{\text{all}} = \{\delta \mid \forall \text{ set } \mathcal{A} \text{ and function } \delta : \Delta^C \to \mathcal{A}\}$.

Typically a decision-maker selects the action that minimizes the expected loss (under the predicted probability). This strategy is formalized by the following definition of Bayes decision-making.

**Definition 1** (Bayes Decision)**.** *Choose any $\ell \in \mathcal{L}_{\text{all}}$ with corresponding action set $\mathcal{A}$ and prediction $\hat{p}$, define the Bayes decision function as $\delta_\ell(\hat{p}(x)) = \arg\inf_{a \in \mathcal{A}} \mathbb{E}_{\hat{Y} \sim \hat{p}(x)}[\ell(\hat{Y}, a)]$. For any subset $\mathcal{L} \subset \mathcal{L}_{\text{all}}$ denote the set of all Bayes decision functions as $\mathbb{\Delta}_{\mathcal{L}} := \{\delta_\ell \mid \forall \ell \in \mathcal{L}\}$.*

## 3 Calibration: A Decision-Making Perspective

In our setup, the decision-maker outsources the prediction task to a third-party forecaster, which returns a prediction function $\hat{p}$ (e.g., an ML Prediction API). The decision maker will use $\hat{p}$ and a loss function $\ell$ to make decisions. However, the forecaster does not necessarily know the loss function $\ell$ in advance, and in more challenging cases, needs to serve multiple decision-makers with different loss functions. For example, the prediction function may be trained to optimize a standard objective such as L2 error or log likelihood, then sold to decision-makers as an off-the-shelf solution.

In such a setting, the decision makers may be concerned that the off-the-shelf solution may not perform well according to their loss function. If the forecaster could predict optimally (i.e. $\hat{p}(X) = p^*(X)$ almost surely), then there would be no issue of trust; of course, perfect predictions are usually impossible, so the forecaster needs feasible ways of conveying trust to the decision makers. To mitigate concerns about the performance of the prediction function, the forecaster might aim to offer performance guarantees applicable to decision makers whose loss functions come from class of losses $\mathcal{L} \subset \mathcal{L}_{\text{all}}$.

### 3.1 Decision Calibration

First and foremost, a decision maker wants assurance that the Bayes decision rule $\delta_\ell$ with $\hat{p}$ gives low expected loss. Second, the decision maker wants to know how much loss is going to be incurred (before the actions are deployed and outcomes are revealed); the decision maker does not want to incur any additional loss in surprise that she has not prepared for.

To capture these desiderata we formalize a definition based on the following intuition: suppose a decision maker with some loss function $\ell$ considers a decision rule $\delta \in \mathbb{\Delta}_{\text{all}}$ (that may or may not be the Bayes decision rule), the decision maker should be able to correctly compute the expected loss of using $\delta$ to make decisions, as a function of *the predictions* $\hat{p}$.

**Definition 2** (Decision Calibration)**.** *For a set of loss functions $\mathcal{L} \subset \mathcal{L}_{\text{all}}$ and a set of decision rules $\mathbb{\Delta} \subset \mathbb{\Delta}_{\text{all}}$, we say that a prediction $\hat{p}$ is $(\mathcal{L}; \mathbb{\Delta})$-decision calibrated (with respect to $p^*$) if $\forall \ell \in \mathcal{L}$ and $\delta \in \mathbb{\Delta}$ with the same action space $\mathcal{A}$ [2], the computed loss (based on $\hat{p}$) of $\delta$ is the same as the actual*

[2] We require $\ell$ and $\delta$ to have the same action space $\mathcal{A}$ for type check reasons. Eq.(2) only has meaning if the loss $\ell$ and decision rule $\delta$ are associated with the same action space $\mathcal{A}$.

*loss, i.e.*

$$\mathbb{E}_X \mathbb{E}_{\hat{Y}\sim\hat{p}(X)}[\ell(\hat{Y}, \delta(\hat{p}(X)))] = \mathbb{E}_X \mathbb{E}_{Y\sim p^*(X)}[\ell(Y, \delta(\hat{p}(X)))] \tag{1}$$

*In particular, we say $\hat{p}$ is $\mathcal{L}$-decision calibrated if it is $(\mathcal{L}; \mathbb{\Delta}_{\mathcal{L}})$-decision calibrated, where $\mathbb{\Delta}_{\mathcal{L}}$ is the set of all Bayes decision rules for loss functions in $\mathcal{L}$.*

The left hand side of Eq.(2) is the "simulated" loss where the outcome $\hat{Y}$ is hypothetically drawn from the predicted distribution. The decision maker can compute this just by knowing the input features $X$ and without knowing the outcome $Y$. The right hand side of Eq.(2) is the true loss that the decision maker incurs in reality if she uses the decision rule $\delta$. Intuitively, the definition captures the idea that the losses in $\mathcal{L}$ and decision rules in $\mathbb{\Delta}$ do not distinguish between outcomes sampled according to the predicted probability and the true probability; specifically, the definition can be viewed as an instantiation of the framework of *Outcome Indistinguishability* (Dwork et al., 2021).

As a cautionary remark, Eq.(2) should not be mis-interpreted as guarantees about individual decisions; Eq.(2) only looks at the average loss when $X, Y$ is a random draw from the population. Individual guarantees are usually impossible without tools beyond machine learning (Zhao & Ermon, 2021). In addition, Definition 2 does not consider decision rules that can directly depend on $X$, as $\delta \in \mathbb{\Delta}_{\text{all}}$ only depends on $X$ via the predicted probability $\hat{p}(X)$. Studying decision rules that can directly depend on $X$ require tools such as multicalibration (Hébert-Johnson et al., 2018) which are beyond the scope of this paper (see related work).

In Definition 2 we also define the special notion of $\mathcal{L}$-decision calibrated because given a set of loss functions $\mathcal{L}$, we are often only interested in the associated Bayes decision rules $\mathbb{\Delta}_{\mathcal{L}}$, i.e. the set of decision rules that are optimal under *some* loss function. For the rest of the paper we focus on $\mathcal{L}$-decision calibration for simplicity. $\mathcal{L}$-decision calibration can capture the desiderata we discussed above formalized in the following proposition.

**Proposition 1.** *If a prediction function $\hat{p}$ is $\mathcal{L}$-decision calibrated, then it satisfies $\forall \delta' \in \mathbb{\Delta}_{\mathcal{L}}$*

$$\mathbb{E}_X \mathbb{E}_{Y\sim p^*(X)}[\ell(Y, \delta_\ell(\hat{p}(X)))] \leq \mathbb{E}_X \mathbb{E}_{Y\sim p^*(X)}[\ell(Y, \delta'(\hat{p}(X)))] \quad \textit{(Bayes Decision Optimality)}$$

$$\mathbb{E}_X \mathbb{E}_{\hat{Y}\sim\hat{p}(X)}[\ell(\hat{Y}, \delta_\ell(\hat{p}(X)))] = \mathbb{E}_X \mathbb{E}_{Y\sim p^*(X)}[\ell(Y, \delta_\ell(\hat{p}(X)))] \quad \textit{(Accurate loss estimation)}$$

**Bayes Decision Optimality** states that the Bayes decision rule $\delta_\ell$ is not worse than any alternative decision rule $\delta' \in \mathbb{\Delta}_{\mathcal{L}}$. In other words, the decision maker is incentivized to take optimal actions according to their true loss function. That is, using the predictions given by $\hat{p}$, the decision maker cannot improve their actions by using a decision rule $\delta'$ that arises from a different loss function $\ell' \in \mathcal{L}$.

**Accurate loss estimation** states that for the Bayes decision rule $\delta_\ell$, the simulated loss on the left hand side (which the decision maker can compute before the outcomes are revealed) equals the true loss on the right hand side. This ensures that the decision maker knows the expected loss that will be incurred over the distribution of individuals and can prepare for it. This is important because in most prediction tasks, the labels $Y$ are revealed with a significant delay or never revealed. For example, the hospital might be unable to follow up on the true outcome of all of its patients.

In practice, the forecaster chooses some set $\mathcal{L}$ to achieve $\mathcal{L}$-decision calibration, and advertise it to decision makers. A decision makers can then check whether their loss function $\ell$ belongs to the advertised set $\mathcal{L}$. If it does, the decision maker should be confident that the Bayes decision rule $\delta_\ell$ has low loss compared to alternatives in $\mathbb{\Delta}_{\mathcal{L}}$, and they can know in advance the loss that will be incurred.

### 3.2 Decision Calibration Generalizes Existing Notions of Calibration

We show that by varying the choice of loss class $\mathcal{L}$, decision calibration can actually express prior notions of calibration. For example, consider confidence calibration, where among the samples whose the top probability is $\beta$, the top accuracy is indeed $\beta$. Formally, confidence calibration requires that

$$\Pr[Y = \arg\max \hat{p}(X) \mid \max \hat{p}(X) = \beta] = \beta.$$

We show that a prediction function $\hat{p}$ is confidence calibrated if and only if it is $\mathcal{L}_r$-decision calibration, where $\mathcal{L}_r$ is defined by

$$\mathcal{L}_r := \{\ell(y, a) = \mathbb{I}(y \neq a \wedge a \neq \perp) + \beta \cdot \mathbb{I}(a = \perp) \mid a \in \mathcal{Y} \cup \{\perp\}, \forall \beta \in [0, 1]\}$$

| Existing Calibration Definitions | Associated Loss Functions |
|---|---|
| Confidence Calibration (Guo et al., 2017)<br>$\Pr[Y = \arg\max \hat{p}(X) \mid \max \hat{p}(X) = \beta] = \beta$<br>$\forall \beta \in [0,1]$ | $\mathcal{L}_r := \{\ell(y,a;\beta) \mid \forall \beta \in [0,1]\}$<br>where $\ell(y,a;\beta) := \mathbb{I}(y \neq a \wedge a \neq \perp) + \beta \cdot \mathbb{I}(a = \perp)$<br>$a \in \mathcal{Y} \cup \{\perp\}$ |
| Classwise Calibration (Kull et al., 2019)<br>$\mathbb{E}[Y_c \mid \hat{p}_c(X) = \beta] = \beta, \forall c \in [C], \forall \beta \in [0,1]$ | $\mathcal{L}_{cr} := \{\ell_c(y,a;\beta_1,\beta_2,c) \mid \forall \beta_1,\beta_2 \in \mathbb{R}, c \in [C]\}$<br>where $\ell(y,a;\beta_1,\beta_2,c) := \mathbb{I}(a = \perp) + \beta_1 \cdot \mathbb{I}(y = c \wedge a = T)$<br>$+\beta_2 \cdot \mathbb{I}(y \neq c \wedge a = F), \quad a \in \{T, F, \perp\}$ |
| Distribution Calibration (Kull & Flach, 2015)<br>$\mathbb{E}[Y \mid \hat{p}(X) = q] = q, \forall q \in \Delta^C$ | $\mathcal{L}_{\text{all}} = \{\ell \mid \forall$ set $\mathcal{A}$ and function $\ell : \mathcal{Y} \times \mathcal{A} \to \mathbb{R}\}$ |

Table 1: A prediction function $\hat{p}$ satisfies the calibration definitions on the left if and only if it satisfies $\mathcal{L}$-decision calibration for the loss function families on the right (Theorem 1).

Intuitively, loss functions in $\mathcal{L}_r$ corresponds to the refrained prediction task: a decision maker chooses between reporting a class label, or reporting "I don't know," denoted $\perp$. She incurs a loss of 0 for correctly predicting the label $y$, a loss of 1 for reporting an incorrect class label, and a loss of $\beta < 1$ for reporting "I don't know". If a decision maker's loss function belong to this simple class of losses $\mathcal{L}_r$, he or she can use a confidence calibrated prediction function $\hat{p}$, because the two desiderata (Bayes decision optimality and accurate loss estimation) in Proposition 1 are true for the decision maker. However, such "refrained prediction" decision tasks only account for a tiny subset of all possible tasks that are interesting to decision makers. Similarly, classwise calibration can be characterized through decision calibration using a class of loss functions that penalizes class-specific false positives and negatives.

In this way, decision calibration clarifies the implications of existing notions of calibration on decision making: relaxed notions of calibration correspond to decision calibration over restricted classes of losses. In general, decision calibration provides a unified view of most existing notions of calibration as the following theorem shows.

**Theorem 1.** *[Decision Calibration Generalizes Existing Notions] For any true distribution $p^*$, and for the loss function sets $\mathcal{L}_r, \mathcal{L}_{cr}$ defined in Table 1, a prediction function $\hat{p}$ is*

- *confidence calibrated iff it is $\mathcal{L}_r$-decision calibrated.*
- *classwise calibrated iff it is $\mathcal{L}_{cr}$-decision calibrated.*
- *distribution calibrated iff it is $\mathcal{L}_{all}$-decision calibrated.*

For proof of this theorem see Appendix C. In Table 1, confidence and classwise calibration are weak notions of calibration; correspondingly the loss function families $\mathcal{L}_r$ and $\mathcal{L}_{\text{cr}}$ are also very restricted.

On the other hand, distribution calibration (i.e. $\mathbb{E}[Y \mid \hat{p}(X) = q] = q, \forall q \in \Delta^C$) is equivalent to $\mathcal{L}_{\text{all}}$-decision calibration. This means that a distribution calibrated predictor guarantees the Bayes decision optimality and accurate loss estimation properties as in Proposition 1 to a decision maker holding any loss functions. Unfortunately, distribution calibration is very challenging to verify or achieve. To understand the challenges, consider certifying whether a given predictor $\hat{p}$ is distribution calibrated. Because we need to reason about the conditional distribution $\mathbb{E}[Y \mid \hat{p}(X) = q]$ for every $q$ that $\hat{p}$ can predict (i.e. the support of $\hat{p}$), the necessary sample complexity grows linearly in the support of $\hat{p}$. Of course, for a trivial predictors that map all inputs $x$ to the same prediction $q_0$ (i.e. $\hat{p}(x) = q_0, \forall x \in \mathcal{X}$) distribution calibration is easy to certify (Widmann et al., 2019), but such predictors have no practical use.

Our characterization of distribution calibration further sheds light on why it is so difficult to achieve. $\mathcal{L}_{\text{all}}$ consists of *all* loss function, including all loss functions $\ell : \mathcal{Y} \times \mathcal{A} \to \mathbb{R}$ whose action space $\mathcal{A}$ contains exponentially many elements (e.g. $|\mathcal{A}| = 2^C$). The corresponding decision rules $\delta \in \mathbb{\Delta}_{\mathcal{L}_{\text{all}}} : \Delta^C \to \mathcal{A}$ may also map $\Delta^C$ to exponentially many possible values. Enforcing Definition 2 for such complex loss functions and decision rules is naturally difficult.

### 3.3 Decision Calibration over Bounded Action Space

In many contexts, directly optimizing for distribution calibration may be overkill. In particular, in most realistic settings, decision makers tend to have a bounded number of possible actions, so the relevant losses come from $\mathcal{L}^K$ for reasonable $K \in \mathbb{N}$. Thus, we consider obtaining decision calibration for all loss functions defined over a bounded number of actions $K$. In the remainder of the

paper, we focus on this restriction of decision calibration to the class of losses with bounded action space; we reiterate the definition of decision calibration for the special case of $\mathcal{L}^K$.

**Definition 3** ($\mathcal{L}^K$-Decision Calibration)**.** *Let $\mathcal{L}^K$ be the set of all loss functions with $K$ actions $\mathcal{L}^K = \{\ell \mid \forall \mathcal{A}, |\mathcal{A}| = K, \forall \ell : \mathcal{Y} \times \mathcal{A} \to \mathbb{R}\}$, we say that a prediction $\hat{p}$ is $\mathcal{L}^K$-decision calibrated (with respect to $p^*$) if $\forall \ell \in \mathcal{L}^K$ and $\delta \in \mathbb{\Delta}_{\mathcal{L}^K}$*

$$\mathbb{E}_X \mathbb{E}_{\hat{Y} \sim \hat{p}(X)}[\ell(\hat{Y}, \delta(\hat{p}(X)))] = \mathbb{E}_X \mathbb{E}_{Y \sim p^*(X)}[\ell(Y, \delta(\hat{p}(X)))] \tag{2}$$

The key result (that we show in Section 4) is that $\mathcal{L}^K$-decision calibration can be efficiently verified and achieved for *all* predictors. Specifically, we show that the sample complexity necessary to learn $\mathcal{L}^K$-decision calibrated predictors depends on the number of actions $K$, not on the support of $\hat{p}$. This is in contrast to the standard approach for establishing distribution calibration, where the sample complexity scales with the support of $\hat{p}$ (which is typically exponentially in the number of classes).

# 4 Achieving Decision Calibration with PAC Guarantees

## 4.1 Approximate $\mathcal{L}^K$ Decision Calibration is Verifiable and Achievable

Decision calibration in Definition 2 usually cannot be achieved perfectly. The definition has to be relaxed to allow for statistical and numerical errors. To meaningfully define approximate calibration we assume that the loss functions are bounded, i.e. no outcome $y \in \mathcal{Y}$ and action $a \in \mathcal{A}$ can incur an infinite loss. In particular, we bound $\ell$ by its 2-norm $\max_a \|\ell(\cdot, a)\|_2 := \max_a \sqrt{\sum_{y \in \mathcal{Y}} \ell(y, a)^2}$. [3]
Now we can proceed to define approximate decision calibration. In particular, we compare the difference between the two sides in Eq.(2) of Definition 2 with the maximum magnitude of the loss function.

**Definition 4** (Approximate decision calibration)**.** *A prediction function $\hat{p}$ is $(\mathcal{L}, \epsilon)$-decision calibrated (with respect to $p^*$) if $\forall \ell \in \mathcal{L}$ and $\delta \in \mathbb{\Delta}_{\mathcal{L}}$*

$$\left| \mathbb{E}[\ell(\hat{Y}, \delta(\hat{p}(X)))] - \mathbb{E}[\ell(Y, \delta(\hat{p}(X)))] \right| \leq \epsilon \sup_{a \in \mathcal{A}} \|\ell(\cdot, a)\|_2 \tag{3}$$

Definition 4 is a relaxation of Definition 2: if a prediction function is $(\mathcal{L}, 0)$-decision calibrated, then it is $\mathcal{L}$-decision calibrated (Definition 2).

The main observation in our paper is that for the set of loss functions with $K$ actions, $(\mathcal{L}^K, \epsilon)$-decision calibration can be verified and achieved with polynomially many samples. In addition, we achieve decision calibration without deteriorating the $L_2$ error $\mathbb{E}[\|\hat{p}(X) - Y\|_2^2]$.

**Theorem 2** (Main Theorem, informal)**.** *There is an algorithm, such that for any predictor $\hat{p}$ and given polynomially (in $K, C, 1/\epsilon$) many samples, can with high probability*

1. *correctly decide if $\hat{p}$ satisfies $(\mathcal{L}^K, \epsilon)$-decision calibration*
2. *output a new predictor $\hat{p}'$ that satisfies $(\mathcal{L}^K, \epsilon)$-decision calibration without degrading the original predictions (in terms of $L_2$ error).*

Note that trivial predictors $\hat{p}(x) \equiv \mathbb{E}[Y], \forall x$ satisfy $(\mathcal{L}^K, 0)$-decision calibration. To maintain a meaningful prediction function we also require "sharpness" which we measure by $L_2$ error. We guarantee that the $L_2$ error of $\hat{p}'$ can only *improve* under post-processing; that is, $\mathbb{E}[\|\hat{p}'(X) - Y\|_2^2] \leq \mathbb{E}[\|\hat{p}(X) - Y\|_2^2]$. In fact, the algorithm works by iteratively updating the predictions to make progress in $L_2$. In addition to $L_2$ error, empirically our recalibration algorithm also slightly improves the likelihood and accuracy. For rest of this section, we propose concrete algorithms that satisfy Theorem 2.

## 4.2 Verification of Decision Calibration

This section focuses on the first part of Theorem 2 where we certify $(\mathcal{L}^K, \epsilon)$-decision calibration. A naive approach use samples to directly estimate (by replacing all expectations with empirical sample

[3] The choice of 2-norm is for convenience. All $p$-norms are equivalent up to a multiplicative factor polynomial in $C$, so our main theorem (Theorem 2) still hold for any $p$-norms up to the polynomial factor.

averages)

$$\sup_{\delta \in \mathbb{\Delta}_{\mathcal{L}^K}} \sup_{\ell \in \mathcal{L}^K} \left| \mathbb{E}[\ell(\hat{Y}, \delta(\hat{p}(X)))] - \mathbb{E}[\ell(Y, \delta(\hat{p}(X)))] \right| / \left( \sup_{a \in \mathcal{A}} \|\ell(\cdot, a)\|_2 \right) \tag{4}$$

and compare it with $\epsilon$. Even though it might be possible to upper bound the estimation error for Eq.(4) directly, the analysis would be complex because of the multiple $\sup$ in the equation. To gain deeper insight into the problem and simplify its analysis, we will make several observations to transform this complex optimization problem to a simpler problem that resembles linear classification.

**Observation 1.** The first observation is that we do not have to take the supremum over $\mathcal{L}^K$ because for any choice of $\delta \in \mathbb{\Delta}$ by simple calculations we have

$$\sup_{\ell \in \mathcal{L}^K} \frac{\left| \mathbb{E}[\ell(\hat{Y}, \delta(\hat{p}(X)))] - \mathbb{E}[\ell(Y, \delta(\hat{p}(X)))] \right|}{\sup_{a \in \mathcal{A}} \|\ell(\cdot, a)\|_2} = \sum_{a=1}^{K} \left\| \mathbb{E}[(\hat{Y} - Y)\mathbb{I}(\delta(\hat{p}(X)) = a)] \right\|_2 \tag{5}$$

This statement is formally proved as part of the proof for Proposition 3 in Appendix C.3. Intuitively on the right hand side, $\delta$ partitions the probabilities $\Delta^C$ based on the optimal decision $\Delta_1 := \{q \in \Delta^C, \mathbb{I}(\delta(q) = 1)\}, \cdots, \Delta_K := \{q \in \Delta^C, \mathbb{I}(\delta(q) = K)\}$. For each partition $\Delta_k$ we measure the difference between predicted label and true label *on average*, i.e. $\mathbb{E}[(\hat{Y} - Y)\mathbb{I}(\hat{p}(X) \in \Delta_k)]$.

**Observation 2.** We observe that the partitions of $\Delta^C$ are defined by linear classification boundaries. Formally, we introduce a new notation for the linear multi-class classification functions

$$B^K = \left\{ b_w \mid \forall w \in \mathbb{R}^{K \times C} \right\} \qquad \text{where } b_w(q, a) = \mathbb{I}(a = \arg \sup_{a' \in [K]} \langle q, w_{a'} \rangle) \tag{6}$$

Note that this new classification task is a tool to aid in understanding decision calibration, and bears no relationship with the original prediction task (predicting $Y$ from $X$). Intuitively $w$ defines the weights of a linear classifier; given input features $q \in \Delta^C$ and a candidate class $a$, $b_w$ outputs an indicator: $b_w(q, a) = 1$ if the optimal decision of $q$ is equal to $a$ and 0 otherwise.

The following equality draws the connection between Eq.(5) and linear classification. The proof is simply a translation from the original notations to the new notations.

$$\sup_{\delta \in \mathbb{\Delta}_{\mathcal{L}^K}} \sum_{a=1}^{K} \left\| \mathbb{E}[(\hat{Y} - Y)\mathbb{I}(\delta(\hat{p}(X))) = a)] \right\|_2 = \sup_{b \in B^K} \sum_{a=1}^{K} \left\| \mathbb{E}[(\hat{Y} - Y) b(\hat{p}(X), a)] \right\|_2 \tag{7}$$

The final outcome of our derivations is the following proposition (Proof in Appendix C.3)

**Proposition 2.** *A predictor $\hat{p}$ satisfies $(\mathcal{L}^K, \epsilon)$-decision calibration if and only if*

$$\sup_{b \in B^K} \sum_{a=1}^{K} \left\| \mathbb{E}[(\hat{Y} - Y) b(\hat{p}(X), a)] \right\|_2 \leq \epsilon \tag{8}$$

In words, $\hat{p}$ satisfies decision calibration if and only if there is no linear classification function that can partition $\Delta^C$ into $K$ parts, such that the average difference $\hat{Y} - Y$ (or equivalently $\hat{p}(X) - p^*(X)$) in each partition is large. Theorem 2.1 follows naturally because $B^K$ has low Radamacher complexity, so the left hand side of Eq.(8) can be accurately estimated with polynomially many samples. For a formal statement and proof see Appendix C.3.

The remaining problem is computation. With unlimited compute, we can upper bound Eq.(8) by brute force search over $B^K$; in practice, we use a surrogate objective optimizable with gradient descent. This is the topic of Section 4.4.

### 4.3 Recalibration Algorithm

This section discusses the second part of Theorem 2 where we design a post-processing recalibration algorithm. The algorithm is based on the following intuition, inspired by (Hébert-Johnson et al., 2018): given a predictor $\hat{p}$ we find the worst $b \in B^K$ that violates Eq.(8) (line 3 of Algorithm 1); then,

we make an update to $\hat{p}$ to minimize the violation of Eq.(8) for the worst $b$ (line 4,5 of Algorithm 1). This process is be repeated until we get a $(B^K, \epsilon)$-decision calibrated prediction (line 2). The sketch of the algorithm is shown in Algorithm 1 and the detailed algorithm is in the Appendix.

**Algorithm 1:** Recalibration algorithm to achieve $\mathcal{L}^K$ decision calibration.

1 Input current prediction function $\hat{p}$, tolerance $\epsilon$. Initialize $\hat{p}^{(0)} = \hat{p}$;
2 **for** $t = 1, 2, \cdots, T$ *until output* $\hat{p}^{(T)}$ *when it satisfies Eq.(8)* **do**
3 Find $b \in B^K$ that maximizes $\sum_{a=1}^{K} \left\| \mathbb{E}[(Y - \hat{p}^{(t-1)}(X))b(\hat{p}^{(t-1)}(X), a) \right\|$ ;
4 Compute the adjustments $d_a = \mathbb{E}[(Y - \hat{p}^{(t-1)}(X))b(\hat{p}^{(t-1)}(X), a)]/\mathbb{E}[b(\hat{p}^{(t-1)}(X), a)]$ ;
5 Set $\hat{p}^{(t)} : x \mapsto \hat{p}^{(t-1)}(x) + \sum_{a=1}^{K} b(\hat{p}^{(t-1)}(x), a) \cdot d_a$ (projecting onto $[0, 1]$ if necessary) ;
6 **end**

Given a dataset with $N$ samples, the expectations in Algorithm 1 are replaced with empirical averages. The following theorem demonstrates that Algorithm 1 satisfies the conditions stated in Theorem 2.

**Theorem 2.2.** *Given any input $\hat{p}$ and tolerance $\epsilon$, Algorithm 1 terminates in $O(K/\epsilon^2)$ iterations. For any $\lambda > 0$, given $O(poly(K, C, \log(1/\delta), \lambda))$ samples, with $1 - \delta$ probability Algorithm 1 outputs a $(\mathcal{L}^K, \epsilon + \lambda)$-decision calibrated prediction function $\hat{p}'$ that satisfies $\mathbb{E}[\|\hat{p}'(X) - Y\|_2^2] \leq \mathbb{E}[\|\hat{p}(X) - Y\|_2^2] + \lambda$.*

The theorem demonstrates that if we can solve the inner search problem over $B^K$, then we can obtain decision calibrated predictions efficiently in samples and computation.

### 4.4 Relaxation of Decision Calibration for Computational Efficiency

We complete the discussion by addressing the open computational question. Directly optimizing over $B^K$ is difficult, so we instead define the softmax relaxation.

$$\bar{B}^K = \left\{ \bar{b}_w \mid \forall w \in \mathbb{R}^{K \times C} \right\} \qquad \text{where } \bar{b}_w(q, a) = \frac{e^{\langle q, w_a \rangle}}{\sum_{a'} e^{\langle q, w_{a'} \rangle}}$$

The key motivation behind this relaxation is that $\bar{b}_w \in \bar{B}^K$ is now differentiable in $w$, so we can optimize over $\bar{B}^K$ using gradient descent. Correspondingly some technical details in Algorithm 1 change to accommodate soft partitions; we address these modifications in Appendix A and show that after these modifications Theorem 2.2 still holds. Intuitively, the main reason that we can still achieve decision calibration with softmax relaxation is because $B^K$ is a subset of the closure of $\bar{B}^K$. Therefore, compared to Eq.(8), we enforce a slightly stronger condition with the softmax relaxation. This can be formalized in the following proposition.

**Proposition 3.** *A prediction function $\hat{p}$ is $(\mathcal{L}^K, \epsilon)$-decision calibrated if it satisfies*

$$\sup_{\bar{b} \in \bar{B}^K} \sum_{a=1}^{K} \left\| \mathbb{E}[(\hat{Y} - Y)\bar{b}(\hat{p}(X), a)] \right\|_2 \leq \epsilon : \tag{9}$$

We remark that unlike Proposition 2 (which is an "if and only if" statement), Proposition 3 is a "if" statement. This is because Eq.(9) implies Eq.(8) but not vice versa.

## 5 Empirical Evaluation

### 5.1 Skin Legion Classification

This experiment materializes our motivating example in the introduction. We aim to show on a real medical prediction dataset, our recalibration algorithm improves both the decision loss and reduces the decision loss estimation error. For the estimation error, as in Definition 4 for any loss function $\ell$ and corresponding Bayes decision rule $\delta_\ell$ we measure

$$\text{loss gap} := \left| \mathbb{E}[\ell(\hat{Y}, \delta_\ell(\hat{p}(X)))] - \mathbb{E}[\ell(Y, \delta_\ell(\hat{p}(X)))] \right| / \sup_{a \in \mathcal{A}} \|\ell(\cdot, a)\|_2 \tag{10}$$

In addition to the loss function in Figure 1 (which is motivated by medical domain knowledge), we also consider a set of 500 random loss functions where for each $y \in \mathcal{Y}, a \in \mathcal{A}$, $\ell(y, a) \sim \text{Normal}(0, 1)$, and report both the average loss gap and the maximum loss gap across the loss functions.

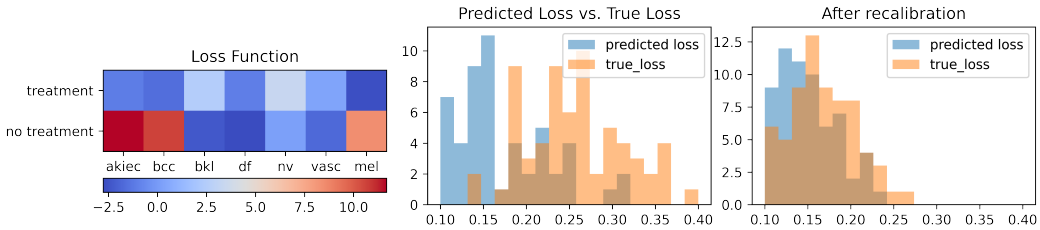


Figure 1: An illustrative example of accurate loss estimation on the HAM10000 dataset. The predictor predicts the probability of 7 skin illness categories (akiec, bcc, ...., mel); the hospital decides between two actions (treatment vs. no treatment). **Left** An example loss function (details not important). Blue indicates low loss and red indicates high loss. For the malignant conditions such as 'akiec' and 'mel', no treatment has high loss (red); for the benign conditions such as 'nv', (unnecessary) treatment has moderate loss. **Middle** Histogram of predicted loss vs. the true loss. The predicted loss is the loss the hospital expected to incur assuming the predicted probabilities are correct under the Bayes decision rule.The true loss is the loss the hospital actually incurs (after ground-truth outcomes are revealed). We plot the histogram from random train/test splits of the dataset, and observe that the true loss is generally much greater than the predicted loss. Because the hospital might incur loss it didn't expect or prepare for, the hospital cannot trust the prediction function. **Right** After applying our recalibration algorithm, the true loss almost matches the predicted loss.

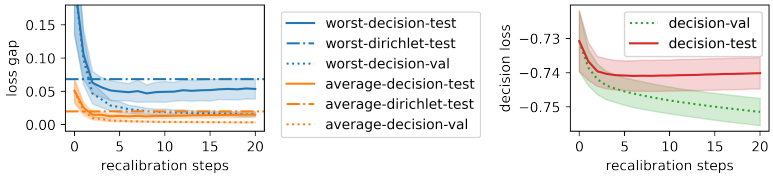


Figure 2: Calibration improves decision loss and its estimation on the HAM10000 skin legion classification dataset. **Left** The gap between the predicted decision loss and the true decision loss in Eq.(10) on a set of randomly sampled loss functions. We plot both the average gap (orange) and the worst gap (blue) out of all the loss functions. The dotted lines are the validation set performance, and solid lines are the test set performance. With more recalibration steps in Algorithm 1, both the average gap and the worst gap improves. The improvements are greater than Dirichlet recalibration (dashed line) **Right** The decision loss (averaged across all random loss functions). With more recalibration steps the decision loss also improves slightly. Dirichlet calibration (dash-dot line) also improves the loss gap for this dataset, but not as much as decision calibration. Note that Dirichlet calibration only has to be applied once, so its performance is a constant instead of a function of the recalibration steps.

**Setup** We use the HAM10000 dataset (Tschandl et al., 2018). We partition the dataset into train/validation/test sets, where approximately 15% of the data are used for validation, while 10% are used for the test set. We use the train set to learn the baseline classifier $\hat{p}$, validation set to recalibrate, and the test set to measure final performance. For modeling we use the densenet-121 architecture (Huang et al., 2017), which achieves around 90% accuracy.

For our method we use Algorithm 2 in Appendix A (which is a small extension of Algorithm 1 explained in Section 4.4). We compare with temperature scaling (Guo et al., 2017) and Dirichlet calibration (Kull et al., 2019). In this experiment, we first apply temperature scaling then apply other methods (decision calibration or Dirichlet calibration). The temperature scaling baseline corresponds to 0 decision recalibration steps in Figure 2.

**Results** The results are shown in Figure 2. For these experiments we set the number of actions $K = 3$. For other choices we obtain qualitatively similar results in Appendix B. The main observation is that decision recalibration improves the loss gap in Eq.(10) and slightly improves the decision loss. We also observe that our recalibration algorithm slightly improves top-1 accuracy (the average improvement is $0.40 \pm 0.08\%$) and L2 loss (the average decrease is $0.010 \pm 0.001$) on the test set.

### 5.2 Imagenet Classification

We stress test our algorithm on Imagenet. The aim is to show that even with deeply optimized classifiers (such as inception-v3 and resnet) that are tailor made for the Imagenet benchmark and a large number of classes, our recalibration algorithm can improve the loss gap in Eq.(10).

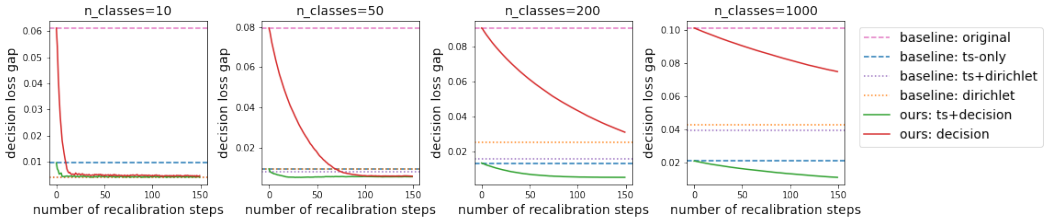


Figure 3: Calibration improves decision loss estimation gap on Imagenet-1000 with inception-v3 on the test set. The meaning of the plot is identical to Figure 2. From left to right we vary the number of classification categories $C$ from 10 to 1000. Our algorithm reduces the gap between predicted loss and true loss in Eq.(10) even up to 1000 classes, although more classes require more iterations and are more prone to over-fitting. Dirichlet calibration is less scalable than our method, and has worse loss estimation with more classes. Using temperature scaling (ts) in additional to decision calibration drastically reduces the number of iterations required for decision calibration to converge.

**Setup** The setup and baselines are identical to the HAM10000 experiment with two differences: we use pretrained models provided by pytorch, and among the 50000 standard validation samples, we use 40000 samples for recalibration and 10000 samples for testing. Similar to the previous experiment, we randomly generate a set of 500 loss functions from normal distributions.

We additional test the performance of decision calibration under distribution drift on the Imagenet-C benchmark (Hendrycks & Dietterich, 2019). We partition the Imagenet validation data into a recalibration set and a test set. We train decision calibration on the recalibration set (with no corruption), and measure performance on the test set (with corruption).

**Results** The results are shown in Figure 3 with additional plots in Appendix B. Decision calibration can generalize to a larger number of classes, and still provides some (albeit smaller) benefits with 1000 classes. Recalibration does not hurt accuracy and L2 error, as we observe that both improve by a modest amount (on average by $0.30\%$ and $0.00173$ respectively). We contrast decision calibration with Dirichlet calibration (Kull et al., 2019). Dirichlet calibration also reduces the loss gap when the number of classes is small (e.g. 10 classes), but is less scaleble than decision recalibration. With more classes its performance degrades much more than decision calibration. In the appendix Figure 6 and Figure 7, we further show that even under distribution drift (on the Imagenet-C) benchmark, decision calibration can improve both decision loss and decision loss estimation.

## 6 Related Work

**Calibration** Early calibration research focus on binary classification (Brier, 1950; Dawid, 1984). For multiclass classification, the strongest definition is distribution (strong) calibration (Kull & Flach, 2015; Song et al., 2019) but is hindered by sample complexity. Weaker notions such as confidence (weak) calibration (Platt et al., 1999; Guo et al., 2017), class-wise calibration (Kull et al., 2019) or average calibration average calibration (Kuleshov et al., 2018) are more commonly used in practice. To unify these notions, (Widmann et al., 2019) proposes $\mathcal{F}$-calibration but lacks detailed guidance on which notions to use.

**Individual calibration** Our paper discusses the *average* decision loss over the population $X$. A stronger requirement is to guarantee the loss for each individual decision. Usually individual guarantees are near-impossible (Barber et al., 2019) and are only achievable with hedging (Zhao & Ermon, 2021) or randomization (Zhao et al., 2020).

**Multi-calibration and Outcome Indistinguishability.** Calibration have been the focus of many works on fairness, starting with (Kleinberg et al., 2016; Pleiss et al., 2017). Multi-calibration has emerged as a noteworthy notion of fairness (Hébert-Johnson et al., 2018; Kim et al., 2019; Dwork et al., 2019; Shabat et al., 2020; Jung et al., 2020; Dwork et al., 2021) because it goes beyond "protected" groups, and guarantees calibration for any group that is identifiable within some computational bound. Recently, (Dwork et al., 2021) generalizes multicalibration to outcome indistinguishability (OI). Decision calibration is a special form of OI.

## 7 Acknowledgements

SZ is supported in part by a JP Morgan fellowship and a Qualcomm innovation fellowship. MPK is supported by the Miller Institute for Basic Research in Science and the Simons Collaboration on the Theory of Algorithmic Fairness. TM acknowledges support of Google Faculty Award and NSF IIS 2045685. SE acknowledges support by NSF(#1651565, #1522054, #1733686), ONR (N000141912145), AFOSR (FA95501910024), ARO (W911NF-21-1-0125) and Sloan Fellowship.

## References

Rina Foygel Barber, Emmanuel J Candes, Aaditya Ramdas, and Ryan J Tibshirani. The limits of distribution-free conditional predictive inference. *arXiv preprint arXiv:1903.04684*, 2019.

Glenn W Brier. Verification of forecasts expressed in terms of probability. *Monthly weather review*, 78(1):1–3, 1950.

A Philip Dawid. Present position and potential developments: Some personal views statistical theory the prequential approach. *Journal of the Royal Statistical Society: Series A (General)*, 147(2): 278–290, 1984.

Cynthia Dwork, Michael P Kim, Omer Reingold, Guy N Rothblum, and Gal Yona. Learning from outcomes: Evidence-based rankings. In *2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 106–125. IEEE, 2019.

Cynthia Dwork, Michael P Kim, Omer Reingold, Guy N Rothblum, and Gal Yona. Outcome indistinguishability. *STOC*, 2021.

Yuguang Fang, Kenneth A Loparo, and Xiangbo Feng. Inequalities for the trace of matrix product. *IEEE Transactions on Automatic Control*, 39(12):2489–2490, 1994.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, pp. 1321–1330. PMLR, 2017.

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

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019.

Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.

Christopher Jung, Changhwa Lee, Mallesh M Pai, Aaron Roth, and Rakesh Vohra. Moment multicalibration for uncertainty estimation. *arXiv preprint arXiv:2008.08037*, 2020.

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*, pp. 247–254, 2019.

Jon Kleinberg, Sendhil Mullainathan, and Manish Raghavan. Inherent trade-offs in the fair determination of risk scores. *arXiv preprint arXiv:1609.05807*, 2016.

Volodymyr Kuleshov, Nathan Fenner, and Stefano Ermon. Accurate uncertainties for deep learning using calibrated regression. In *International Conference on Machine Learning*, pp. 2796–2804. PMLR, 2018.

Meelis Kull and Peter Flach. Novel decompositions of proper scoring rules for classification: Score adjustment as precursor to calibration. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 68–85. Springer, 2015.

Meelis Kull, Miquel Perello-Nieto, Markus Kängsepp, Hao Song, Peter Flach, et al. Beyond temperature scaling: Obtaining well-calibrated multiclass probabilities with dirichlet calibration. *arXiv preprint arXiv:1910.12656*, 2019.

Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2018.

John Platt et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74, 1999.

Geoff Pleiss, Manish Raghavan, Felix Wu, Jon Kleinberg, and Kilian Q Weinberger. On fairness and calibration. In *Advances in Neural Information Processing Systems*, pp. 5680–5689, 2017.

Eliran Shabat, Lee Cohen, and Yishay Mansour. Sample complexity of uniform convergence for multicalibration. *NeurIPS*, 2020.

Hao Song, Tom Diethe, Meelis Kull, and Peter Flach. Distribution calibration for regression. In *International Conference on Machine Learning*, pp. 5897–5906. PMLR, 2019.

Philipp Tschandl, Cliff Rosendahl, and Harald Kittler. The ham10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. *Scientific data*, 5(1):1–9, 2018.

David Widmann, Fredrik Lindsten, and Dave Zachariah. Calibration tests in multi-class classification: A unifying framework. *arXiv preprint arXiv:1910.11385*, 2019.

Shengjia Zhao and Stefano Ermon. Right decisions from wrong predictions: A mechanism design alternative to individual calibration. In *International Conference on Artificial Intelligence and Statistics*, pp. 2683–2691. PMLR, 2021.

Shengjia Zhao, Tengyu Ma, and Stefano Ermon. Individual calibration with randomized forecasting. *arXiv preprint arXiv:2006.10288*, 2020.

## A Relaxations to Decision Calibration

We define the algorithm that corresponds to Algorithm 1 but for softmax relaxed functions. Before defining our algorithm at each iteration $t$ we first lighten our notation with a shorthand $b_a(X) = b(\hat{p}^{(t-1)}(X), a)$ (at different iteration $t$, $b_a$ denotes different functions), and $b(X)$ is the vector of $(b_1(X), \cdots, b_K(X))$.

**Algorithm 2:** Recalibration Algorithm to achieve decision calibration.

1 Input current prediction function $\hat{p}$, a dataset $\mathcal{D} = \{(x_1, y_1), \cdots, (x_M, y_M)\}$ tolerance $\epsilon$ ;
2 Initialize $\hat{p}^{(0)} = \hat{p}$, $v^{(0)} = +\infty$;
3 **for** $t = 1, 2, \cdots$ *until* $v^{(t-1)} < \epsilon^2/K$ **do**
4 $\quad v^{(t)}, b^{(t)} = \sup, \arg\sup_{b \in \bar{B}^K} \sum_{a=1}^{K} \left\| \mathbb{E}[(Y - \hat{p}^{(t-1)}(X))b_a(X)] \right\|^2$ ;
5 $\quad$ Compute $D \in \mathbb{R}^{K \times K}$ where $D_{aa'} = \mathbb{E}[b_a^{(t)}(X) b_{a'}^{(t)}(X)]$ ;
6 $\quad$ Compute $R \in \mathbb{R}^{K \times C}$ where $R_a = \mathbb{E}[(Y - \hat{p}^{(t-1)}(X))b_a^{(t)}(X)]$ ;
7 $\quad$ Set $\hat{p}^{(t)} : x \mapsto \pi(\hat{p}^{(t-1)}(x) + R^T D^{-1} b^{(t)}(x))$ where $\pi$ is the normalization projection;
8 **end**
9 Output $\hat{p}^{(T)}$ where $T$ is the number of iterations

For the intuition of the algorithm, consider the $t$-th iteration where the current prediction function is $\hat{p}^{(t-1)}$. On line 4 we find the worst case $b^{(t)}$ that maximizes the "error" $\sum_{a=1}^{K} \left\| \mathbb{E}[(Y - \hat{p}^{t-1}(X))b_a^{(t)}(X)] \right\|^2$, and on line 5-7 we make the adjustment $\hat{p}^{(t-1)} \to \hat{p}^{(t)}$ to minimize this error $\sum_{a=1}^{K} \left\| \mathbb{E}[(Y - \hat{p}^{t}(X))b_a^{(t)}(X)] \right\|^2$. In particular, the adjustment we aim to find on line 5-7 (which we denote by $U \in \mathbb{R}^{C \times K}$) should satisfy the following: if we let

$$\hat{p}^{(t)}(X) = \hat{p}^{(t-1)}(X) + U b^{(t)}(X)$$

we can minimize

$$L(U) := \sum_{a=1}^{K} \left\| \mathbb{E}[(Y - \hat{p}^{(t)}(X))b_a^{(t)}(X)] \right\|^2$$

We make some simple algebra manipulations on $L$ to get

$$\begin{aligned} L(U) &= \sum_{a=1}^{K} \left\| \mathbb{E}[(Y - \hat{p}^{(t-1)}(X))b_a^{(t)}(X) - U b^{(t)}(X) b_a^{(t)}(X)] \right\|^2 \\ &= \sum_{a=1}^{K} \left\| R_a - (DU^T)_a \right\|^2 = \left\| R - DU^T \right\|^2 \end{aligned}$$

Suppose $D$ is invertible, then the optimum of the objective is

$$U^* := \arg\inf L(U) = R^T D^{-1}, \quad L(U^*) = 0$$

When $D$ is not invertible we can use the pseudo-inverse, though we observe in the experiments that $D$ is always invertible.

For the relaxed algorithm we also have a theorem that is equivalent to Theorem 2.2. The statement of the theorem is identical; the proof is also essentially the same except for the use of some new technical tools.

**Theorem 2.2'.** *Algorithm 2 terminates in $O(K/\epsilon^2)$ iterations. For any $\lambda > 0$, given $O(poly(K, C, \log(1/\delta), \lambda))$ samples, with $1 - \delta$ probability Algorithm 1 outputs a $(\mathcal{L}^K, \epsilon + \lambda)$-decision calibrated prediction function $\hat{p}'$ that satisfies $\mathbb{E}[\|\hat{p}'(X) - Y\|_2^2] \leq \mathbb{E}[\|\hat{p}(X) - Y\|_2^2] + \lambda$.*

## B Additional Experiment Details and Results

Additional experiments are shown in Figure 4 and Figure 5. The observations are similar to those in the main paper.

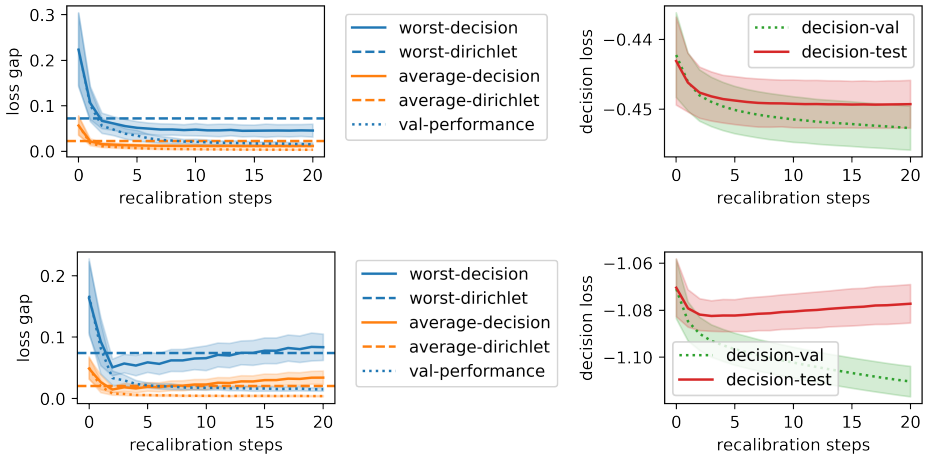


Figure 4: Additional Results on the HAM10000 for 2 and 5 actions. The observations are similar to Figure 2 even though overfitting happens sooner with 5 actions

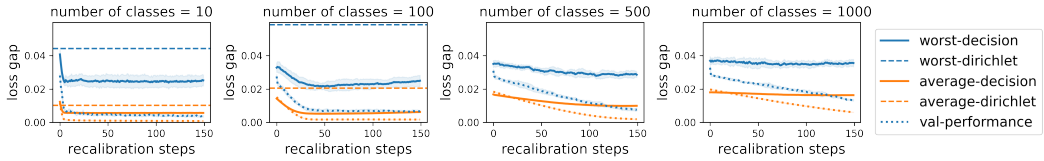


Figure 5: Additional results on the resnet18. The observations are similar to Figure 3: decision recalibration improves the loss gap.

### B.1 Robustness to Distribution Shift

To further test whether decision calibration improves loss under distribution drift, we use the Imagenet-C benchmark (Hendrycks & Dietterich, 2019).

## C Proofs

### C.1 Equivalence between Decision Calibration and Existing Notions of Calibration

**Theorem 1.** *[Decision Calibration Generalizes Existing Notions] For any true distribution $p^*$, and for the loss function sets $\mathcal{L}_r, \mathcal{L}_{cr}$ defined in Table 1, a prediction function $\hat{p}$ is*

- *confidence calibrated iff it is $\mathcal{L}_r$-decision calibrated.*
- *classwise calibrated iff it is $\mathcal{L}_{cr}$-decision calibrated.*
- *distribution calibrated iff it is $\mathcal{L}_{all}$-decision calibrated.*

*Proof of Theorem 1, part 1.* Before the proof we first need a technical Lemma

**Lemma 1.** *For any pair of random variables $U, V$, $\mathbb{E}[U \mid V] = 0$ almost surely if and only if $\forall c \in \mathbb{R}, \mathbb{E}[U\mathbb{I}(V > c)] = 0$.*

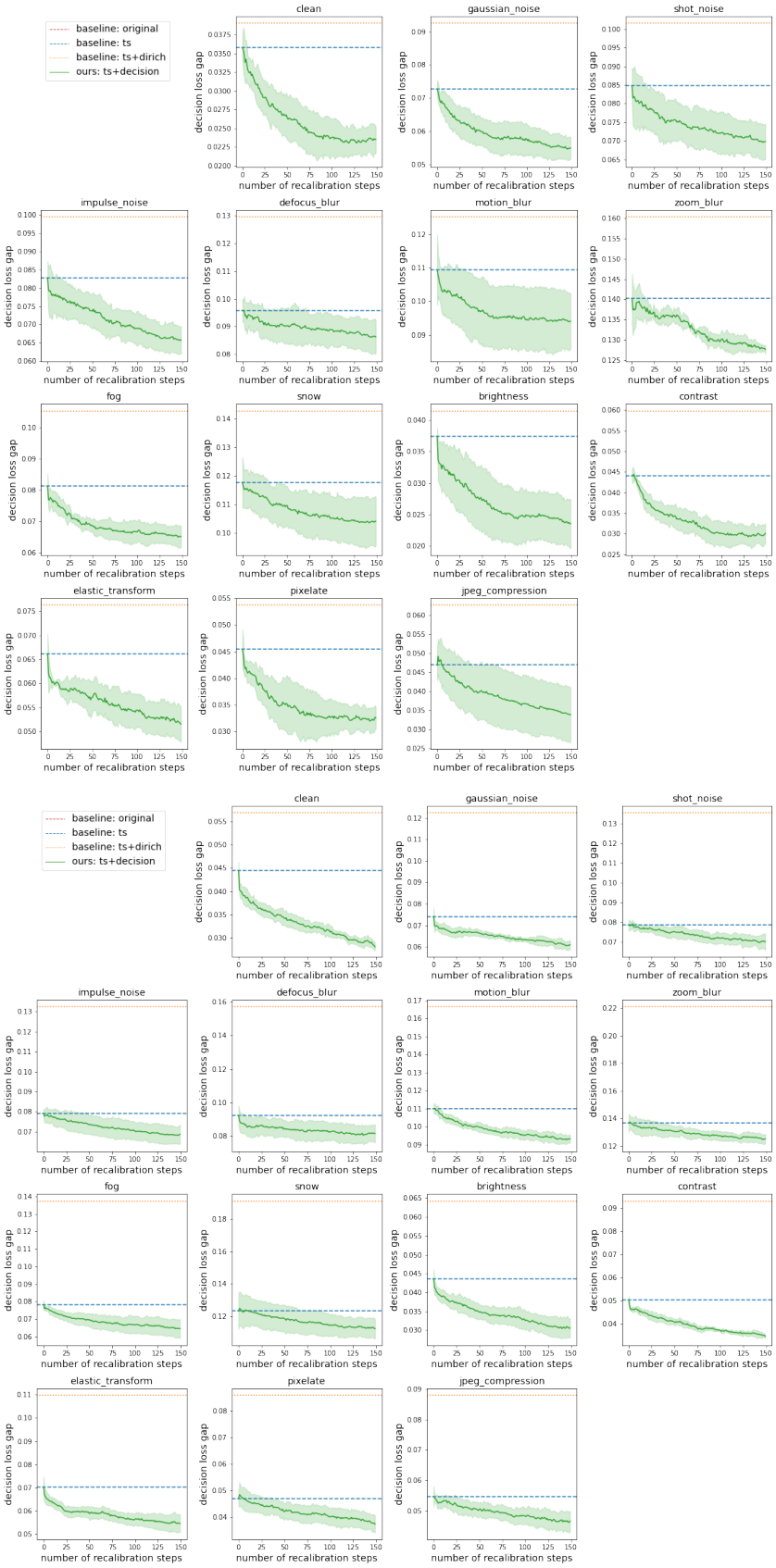


Figure 6: Comparing the maximum difference between predicted loss and actual loss under distribution drifts. The top panel plots the results on 200 classes, and the bottom panel plots the results on 1000 classes.

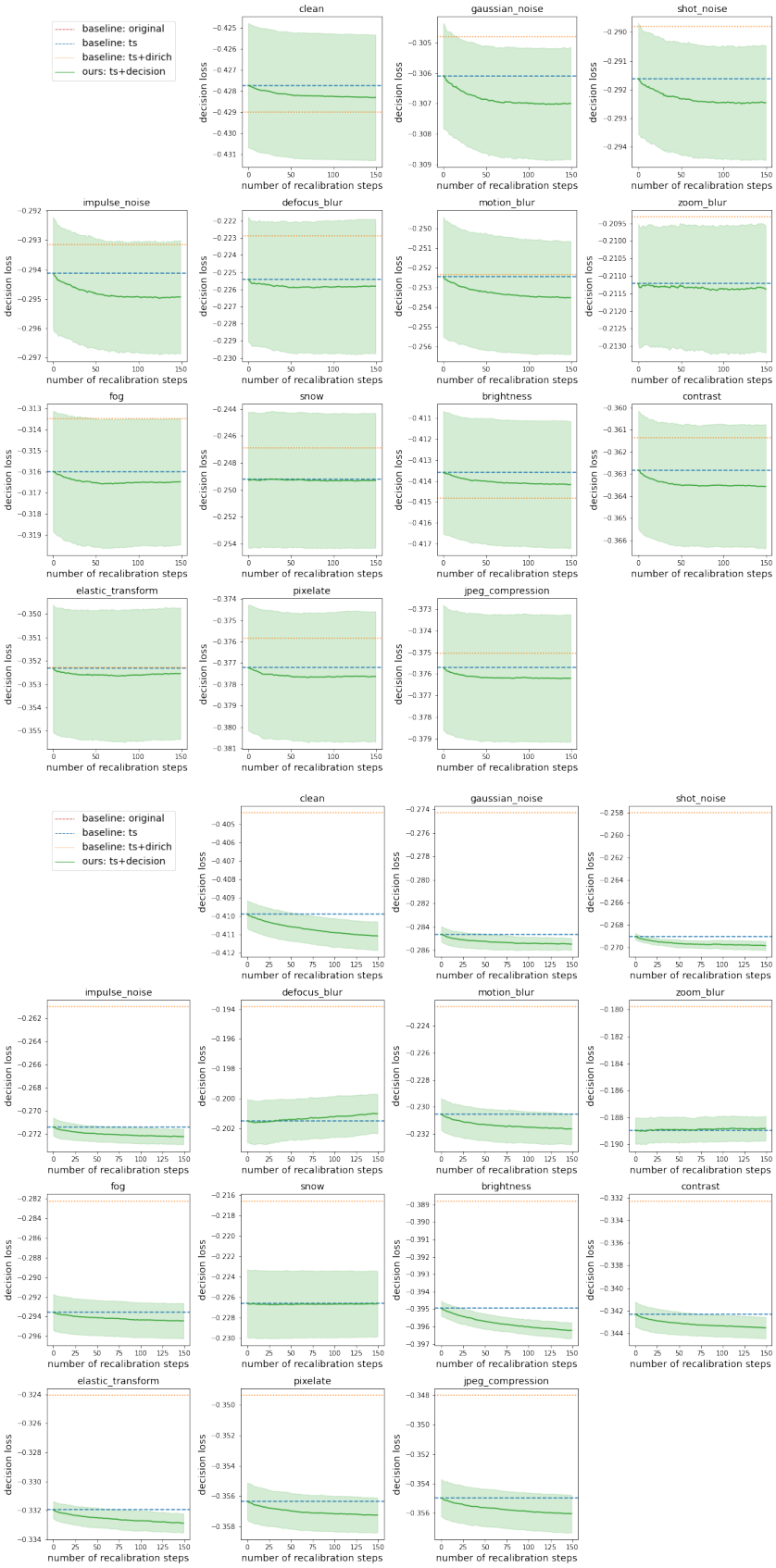


Figure 7: Comparing the decision loss under distribution drifts. The top panel plots the results on 200 classes, and the bottom panel plots the results on 1000 classes.

**Part 1** When the loss function is $\ell : y, a \mapsto \mathbb{I}(y \neq a \cap a \neq \perp) + \beta \mathbb{I}(a = \perp)$, the Bayes decision is given by

$$\delta_\ell(x) = \begin{cases} \arg\max \hat{p}(x) & \max \hat{p}(x) > 1 - \beta \\ \perp & \text{otherwise} \end{cases}$$

Denote $U = \max \hat{p}(X)$ and $V = \arg\max \hat{p}(X)$. For any pair of loss functions $\ell$ and $\ell'$ parameterized by $\beta$ and $\beta'$ we have

$$\begin{aligned}
&\mathbb{E}[\ell'(Y, \delta_\ell(X))] - \mathbb{E}[\ell'(\hat{Y}, \delta_\ell(X))] \\
&= \mathbb{E}[(\ell'(Y, \perp) - \ell'(\hat{Y}, \perp))\mathbb{I}(\delta_\ell(X) = \perp)] + \mathbb{E}[(\ell'(Y, \delta_\ell(X)) - \ell'(\hat{Y}, \delta_\ell(X)))\mathbb{I}(\delta_\ell(X) \neq \perp)] && \text{Tower} \\
&= 0 + \mathbb{E}[(p^*_V(X) - \hat{p}_V(X))\mathbb{I}(\max \hat{p}(x) > 1 - \beta)] && \text{Def of } \ell \\
&= \mathbb{E}[(p^*_V(X) - U)\mathbb{I}(U > 1 - \beta)]
\end{aligned}$$

Suppose $\hat{p}$ is confidence calibrated, then almost surely

$$U = \Pr[Y = \arg\max \hat{p}(X) \mid U] = \mathbb{E}[p^*_V(X) \mid U]$$

which implies almost surely $\mathbb{E}[p^*_V(X) - U \mid U] = 0$. By Lemma 1 we can conclude that

$$0 = \mathbb{E}[(p^*_V(X) - U)\mathbb{I}(U > 1 - \beta)] = \mathbb{E}[\ell'(Y, \delta_\ell(X))] - \mathbb{E}[\ell'(\hat{Y}, \delta_\ell(X))]$$

so $\hat{p}$ is $L_r$-weakly calibrated.

Conversely suppose $\hat{p}$ is $L_r$ weakly calibrated, then $\forall \beta \in [0, 1]$, $\mathbb{E}[(p^*_V(X) - U)\mathbb{I}(U > 1 - \beta)] = 0$. By Lemma 1 we can conclude that almost surely

$$0 = \mathbb{E}[p^*_V(X) - U \mid U] = \mathbb{E}[p^*_V(X) \mid U] - U$$

so $\hat{p}$ is confidence calibrated.

**Part 2** For any loss function $\ell : y, a \mapsto \mathbb{I}(a = \perp) + \beta_1 \mathbb{I}(y \neq c \wedge a = T) + \beta_2 \mathbb{I}(y = c \wedge a = F)$ where $\beta_1, \beta_2 > 1$, observe that the Bayes decision for loss function $\ell$ is

$$\delta_\ell(x) = \begin{cases} T & \hat{p}_c(x) > \max(1 - 1/\beta_1, \beta_1/(\beta_1 + \beta_2)) \\ F & \hat{p}_c(x) < \min(1/\beta_2, \beta_1/(\beta_1 + \beta_2)) \\ \perp & \text{otherwise} \end{cases}$$

Choose any pair of numbers $\alpha \geq \gamma$, we can choose $\beta_1, \beta_2$ such that $\alpha := \max(1 - 1/\beta_1, \beta_1/(\beta_1 + \beta_2)), \gamma := \min(1/\beta_2, \beta_1/(\beta_1 + \beta_2))$. For any pair of loss functions $\ell$ and $\ell'$ parameterized by $\beta_1, \beta_2, \beta'_1, \beta'_2$ (with associated threshold $\alpha \geq \gamma, \alpha' \geq \gamma'$) we have

$$\begin{aligned}
&\mathbb{E}[\ell'(Y, \delta_\ell(X)) - \mathbb{E}[\ell'(\hat{Y}, \delta_\ell(X))] \\
&= \mathbb{E}[(\ell'(Y, \delta_\ell(X)) - \ell'(\hat{Y}, \delta_\ell(X)))\mathbb{I}(\delta_\ell(X) = T)] + \mathbb{E}[(\ell'(Y, \delta_\ell(X)) - \ell'(\hat{Y}, \delta_\ell(X)))\mathbb{I}(\delta_\ell(X) = F)] \\
&= \mathbb{E}[(\ell'(Y, T) - \ell'(\hat{Y}, T))\mathbb{I}(\delta_\ell(X) = T)] + \mathbb{E}[(\ell'(Y, F) - \ell'(\hat{Y}, F))\mathbb{I}(\delta_\ell(X) = F)] \\
&= \beta'_1 \mathbb{E}[(\hat{p}_c(X) - p^*_c(X))\mathbb{I}(\hat{p}_c(X) > \alpha)] + \beta'_2 \mathbb{E}[(p^*_c(X) - \hat{p}_c(X))\mathbb{I}(\hat{p}_c(X) < \gamma)]
\end{aligned}$$

Similar to the argument for part 1, suppose $\hat{p}$ is classwise calibrated then $\forall \alpha, \gamma$, $\mathbb{E}[(p^*_c(X) - \hat{p}_c(X))\mathbb{I}(\hat{p}_c(X) > \alpha)] = 0$ and $\mathbb{E}[(p^*_c(X) - \hat{p}_c(X))\mathbb{I}(\hat{p}_c(X) < \gamma)] = 0$; therefore it is $\mathcal{L}_{cr}$-decision calibrated.

Conversely suppose $\hat{p}$ is $\mathcal{L}_{cr}$-decision calibrated, then $\forall \alpha$ we have $\mathbb{E}[(\hat{p}_c(X) - p^*_c(X))\mathbb{I}(\hat{p}_c(X) > \alpha)] = 0$, which implies that $\hat{p}$ is classwise calibrated according to Lemma 1.

**Part 3** Choose the special loss function $\mathcal{A} = \Delta^C$ and $\ell$ as the log loss $\ell : y, a \mapsto -\log a_y$ then the Bayes action can be computed as

$$\delta_\ell(x) = \arg \inf_{a \in \Delta^C} \mathbb{E}_{\hat{Y} \sim \hat{p}(X)}[-\log a_{\hat{Y}}] = \hat{p}(x)$$

Denote $U = \hat{p}(X)$ then let $\mathcal{L}_B$ be the set of all bounded loss functions, i.e. $\mathcal{L}_B = \{\ell, |\ell(y, a)| \leq B\}$

$$\begin{aligned}
&\sup_{\ell' \in \mathcal{L}_B} \mathbb{E}[\ell'(Y, \delta_\ell(X))] - \mathbb{E}[\ell'(\hat{Y}, \delta_\ell(X))] \\
&\sup_{\ell' \in \mathcal{L}_B} \mathbb{E}[\mathbb{E}[\ell'(Y, U) - \ell'(\hat{Y}, U) \mid U]] && \text{Tower}
\end{aligned}$$

$$= B\mathbb{E}[\|\mathbb{E}[p^*(X) - \hat{p}(X) \mid U]\|_1] \qquad \text{Cauchy Schwarz}$$

If $\hat{p}$ satisfies distribution calibration, then $\|\mathbb{E}[p^*(X) - \hat{p}(X) \mid U]\|_1 = 0$ almost surely, which implies that $\hat{p}$ is $\mathcal{L}_B$ decision calibrated. Conversely, if $\hat{p}$ is $\mathcal{L}_B$ decision calibrated, then $\|\mathbb{E}[p^*(X) - \hat{p}(X) \mid U]\|_1 = 0$ almost surely (because if the expectation of a non-negative random variable is zero, the random variable must be zero almost surely), which implies that $\hat{p}$ is distribution calibrated. The theorem follows because $B$ is arbitrarily chosen. □

### C.2 From Decision Calibration to Distribution Calibration

*

*Proof of Proposition* **??**. First we observe from Proposition 2 that if a predictor $\hat{p}$ is $\mathcal{L}^K$-decision calibrated, then for all $a \in [K]$ such that $\{x, \delta_\ell(\hat{p}(x)) = a\}$ has non-zero probability,

$$\mathbb{E}_X \mathbb{E}_{\hat{Y} \sim \hat{p}(X)}[\hat{Y} \mid \delta_\ell(\hat{p}(X)) = a] = \mathbb{E}_X \mathbb{E}_{Y \sim p^*(X)}[Y \mid \delta_\ell(\hat{p}(X)) = a] \tag{11}$$

We now show a simple compression scheme that achieves the properties required for Proposition **??**. Given a loss $\ell \in \mathcal{L}^K$, we compress the predictions along the partitions defined by $\delta_\ell$.

Suppose $\hat{p}$ is $\mathcal{L}^K$-decision calibrated. For a fixed loss $\ell \in \mathcal{L}^K$, consider the following predictor $\hat{p}_\ell$ that arises by compressing $\hat{p}$ according to the optimal decision rule $\delta_\ell$.

$$\hat{p}_\ell(x) = \mathbb{E}_X[\hat{p}(X) \mid \delta_\ell(\hat{p}(X)) = \delta_\ell(\hat{p}(x))]$$

First, note that by averaging over each set $\{x, \delta_\ell(\hat{p}(x) = a)\}$, the support size of $\hat{p}_\ell$ is bounded by at most $K$. Next, we note that by construction and Eq.(11), $\hat{p}_\ell$ is distribution calibrated. To see this, consider each $q \in \Delta^C$ supported by $\hat{p}_\ell$; for each $q$, there is some optimal action $a_q \in [K]$. That is, the sets $\{x : \hat{p}_\ell(x) = q\} = \{x : \delta_\ell(\hat{p}(x)) = a_q\}$ are the same. Distribution calibration follows.

$$\begin{aligned}
\mathbb{E}_X \mathbb{E}_{Y \sim p^*(X)}[Y \mid \hat{p}_\ell(X) = q] &= \mathbb{E}_X \mathbb{E}_{Y \sim p^*(X)}[Y \mid \delta_\ell(\hat{p}(X)) = a_q] \\
&= \mathbb{E}_X \mathbb{E}_{\hat{Y} \sim \hat{p}_\ell(X)}[\hat{Y} \mid \delta_\ell(\hat{p}(X)) = a_q] \\
&= \mathbb{E}_X \mathbb{E}_{\hat{Y} \sim \hat{p}_\ell(X)}[\hat{Y} \mid \hat{p}_\ell(X) = q] \\
&= q
\end{aligned}$$

Finally, it remains to show that the optimal decision rule resulting from $\hat{p}_\ell$ and $\hat{p}$ are the same, pointwise for all $x \in \mathcal{X}$. As an immediate corollary, the expected loss using $\hat{p}_\ell$ and $\hat{p}$ is the same. We show that the decision rule will be preserved by the fact that for each $x \in \mathcal{X}$, the compressed prediction is a convex combination of predictions that gave rise to the same optimal action.

Specifically, consider any $x$ such that $\hat{p}_\ell(x) = q$. By the argument above, there is some action $a_q \in [K]$ that is optimal for all such $x$. Optimality implies that for all $a \in [K]$

$$\langle \ell_{a_q}, \hat{p}(x) \rangle \leq \langle \ell_a, \hat{p}(x) \rangle.$$

Thus, by linearity of expectation, averaging over $\{x : \hat{p}_\ell(x) = q\}$, the optimal action $a_q$ is preserved.

$$\begin{aligned}
\langle \ell_{a_q}, q \rangle &= \langle \ell_{a_q}, \mathbb{E}[\hat{p}(X) \mid \delta_\ell(\hat{p}(X)) = a_q] \rangle \\
&= \mathbb{E}[\langle \ell_{a_q}, \hat{p}(X) \rangle \mid \delta_\ell(\hat{p}(X)) = a_q] \\
&\leq \mathbb{E}[\langle \ell_a, \hat{p}(X) \rangle \mid \delta_\ell(\hat{p}(X)) = a_q] \\
&= \langle \ell_a, \mathbb{E}[\hat{p}(X) \mid \delta_\ell(\hat{p}(X)) = a_q] \rangle \\
&= \langle \ell_a, q \rangle
\end{aligned}$$

Thus, the optimal action is preserved for all $x \in \mathcal{X}$. □

### C.3 Proofs for Section 4

**Proposition 2.** *A predictor $\hat{p}$ satisfies $(\mathcal{L}^K, \epsilon)$-decision calibration if and only if*

$$\sup_{b \in B^K} \sum_{a=1}^{K} \left\| \mathbb{E}[(\hat{Y} - Y) b(\hat{p}(X), a)] \right\|_2 \leq \epsilon \tag{8}$$

*Proof of Proposition 2.* We first introduce a new set of notations to make the proof easier to follow. Because $\mathcal{A} = [K]$ and $\mathcal{Y} \simeq [C]$, a loss function can be uniquely identified with $K$ vectors $\ell_1, \cdots, \ell_K$ where $\ell_{ac} = \ell(c, a)$. Given prediction function $\hat{p} : \mathcal{X} \to \Delta^C$ and the expected loss can be denoted as

$$\mathbb{E}_{\hat{Y} \sim \hat{p}(x)}[\ell(\hat{Y}, a)] = \langle \hat{p}(x), \ell_a \rangle \tag{12}$$

Choose any Bayes decision function $\delta_{\ell'}$ for some loss $\ell' \in \mathcal{L}^K$, as a notation shorthand denote $\delta_{\ell'}(\hat{p}(x)) = \delta_{\ell'}(x)$. We can compute the gap between the left hand side and right hand side of Definition 2 as

$$\begin{aligned}
&\sup_{\ell} \frac{\left|\mathbb{E}_X \mathbb{E}_{\hat{Y} \sim \hat{p}(X)}[\ell(\hat{Y}, \delta_{\ell'}(X))] - \mathbb{E}_X \mathbb{E}_{Y \sim p^*(X)}[\ell(Y, \delta_{\ell'}(X))]\right|}{\sup_a \|\ell(\cdot, a)\|_2} \\
&= \sup_{\ell, \|\ell(\cdot,a)\|_2 \le 1} \left|\mathbb{E}_X \mathbb{E}_{\hat{Y} \sim \hat{p}(X)}[\ell(\hat{Y}, \delta_{\ell'}(X))] - \mathbb{E}_X \mathbb{E}_{Y \sim p^*(X)}[\ell(Y, \delta_{\ell'}(X))]\right| && \text{Normalize} \\
&= \sup_{\ell, \|\ell(\cdot,a)\|_2 \le 1} \left|\mathbb{E}_X \left[\langle \ell_{\delta_{\ell'}(X)}, \hat{p}(X) \rangle\right] - \mathbb{E}_X \left[\langle \ell_{\delta_{\ell'}(X)}, p^*(X) \rangle\right]\right| && \text{Eq.12} \\
&= \sup_{\ell, \|\ell(\cdot,a)\|_2 \le 1} \left|\sum_a \mathbb{E}_X[\langle \hat{p}(X), \ell_a \rangle \mathbb{I}(\delta_{\ell'}(X) = a)] - \sum_a \mathbb{E}_X[\langle p^*(X), \ell_a \rangle \mathbb{I}(\delta_{\ell'}(X) = a)]\right| && \text{Total Probability} \\
&= \sup_{\ell, \|\ell(\cdot,a)\|_2 \le 1} \left|\sum_a \langle \mathbb{E}_X[(p^*(X) - \hat{p}(X)) \mathbb{I}(\delta_{\ell'}(X) = a)], \ell_a \rangle\right| && \text{Linearity} \\
&= \sum_a \|\mathbb{E}_X[(p^*(X) - \hat{p}(X)) \mathbb{I}(\delta_{\ell'}(X) = a)]\|_2 && \text{Cauchy Schwarz}
\end{aligned}$$

Finally we complete the proof by observing that the set of maps

$$\{q, a \mapsto \mathbb{I}(\delta_\ell(q) = a), \ell \in \mathcal{L}^K, q \in \Delta^C\}$$

is the same as the set of maps $B^K$. We do this by establishing a correspondence where $\ell_a = -w_a / \|w_a\|_2$ then

$$\mathbb{I}(\delta_\ell(q) = a) = \mathbb{I}(\arg\inf_{a'} \langle \ell_{a'}, q \rangle = a) = \mathbb{I}(\arg\sup_{a'} \langle w_{a'}, q \rangle = a) = b_w(q, a)$$

□

### C.4 Formal Statements and Proofs for Theorem 2

**Formal Statement of Theorem 2, part I.** We first define a new notation. Given a set of samples $\mathcal{D} = \{(X_1, Y_1), \cdots, (X_N, Y_N)\}$, and for any function $\psi : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ denote $\hat{\mathbb{E}}_{\mathcal{D}}[\psi(X, Y)]$ as the empirical expectation, i.e.

$$\hat{\mathbb{E}}_{\mathcal{D}}[\psi(X, Y)] := \frac{1}{N} \sum_n \psi(X_n, Y_n)$$

**Theorem 2.1** (Formal)**.** *Let $B^K$ be as defined by Eq.(6), for any true distribution over $X, Y$ and any $\hat{p}$, given a set of $N$ samples $\mathcal{D} = \{(X_1, Y_1), \cdots, (X_N, Y_N)\}$, with probability $1 - \delta$ over random draws of $\mathcal{D}$,*

$$\sup_{b \in B^K} \sum_{a=1}^{K} \|\mathbb{E}[(\hat{p}(X) - Y) b(\hat{p}(X), a)]\|_2 - \sum_{a=1}^{K} \left\|\hat{\mathbb{E}}_{\mathcal{D}}[(\hat{p}(X) - Y) b(\hat{p}(X), a)]\right\|_2 \le \tilde{O}\left(\frac{K^{3/2} C}{\sqrt{N}}\right) \tag{13}$$

*where $\tilde{O}$ denotes equal up to constant and logarithmic terms.*

Note that in the theorem $\delta$ does not appear on the RHS of Eq.(13). This is because the bound depends logarithmically on $\delta$.

*Proof of Theorem 2.1.* Before proving this theorem we first need a few uniform convergence Lemmas which we will prove in Appendix C.5.

**Lemma 2.** *Let $B$ by any set of functions $\{b : \Delta^C \to [0,1]\}$ and $U, V$ be any pair of random variables where $U$ takes values in $[-1,1]^C$ and $V$ takes values in $\Delta^C$. Let $\mathcal{D} = \{(U_1, V_1), \cdots, (U_N, V_N)\}$ be an i.i.d. draw of $N$ samples from $U, V$, define the Radamacher complexity of $B$ by*

$$\mathcal{R}_N(B) := \mathbb{E}_{\mathcal{D}, \sigma_i \sim \text{Uniform}(\{-1,1\})} \left[ \sup_{b \in B} \frac{1}{N} \sum_{n=1}^{N} \sigma_i b(V_i) \right]$$

*then for any $\delta > 0$, with probability $1 - \delta$ (under random draws of $\mathcal{D}$), $\forall b \in B$*

$$\|\hat{\mathbb{E}}_{\mathcal{D}}[Ub(V)] - \mathbb{E}[Ub(V)]\|_2 \leq \sqrt{C}\mathcal{R}_N(B) + \sqrt{\frac{2C}{N} \log \frac{2C}{\delta}}$$

**Lemma 3.** *Define the function family*

$$B_a^K = \left\{ b : z \mapsto \mathbb{I}(a = \arg\sup_{a'} \langle z, w_a \rangle), w_a \in \mathbb{R}^C, a = 1, \cdots, K, z \in \Delta^C \right\}$$

$$\bar{B}_a^K = \left\{ b : z \mapsto \frac{e^{\langle z, w_a \rangle}}{\sum_{a'} e^{\langle z, w_{a'} \rangle}}, w_a \in \mathbb{R}^C, a = 1, \cdots, K, z \in \Delta^C \right\}$$

*then $\mathcal{R}_N(B_a^K) = O\left(\sqrt{\frac{CK \log K \log N}{N}}\right)$ and $\mathcal{R}_N(\bar{B}_a^K) = O\left(\left(\frac{K}{N}\right)^{1/4} \log \frac{N}{K}\right)$.*

Proof of the theorem is straight-forward given the above Lemmas. As a notation shorthand denote $U = \hat{p}(X) - Y$. Note that $U$ is a random vector raking values in $[-1,1]^C$. We can rewrite the left hand side of Eq.(13) as

$$\sup_{b \in B^K} \sum_a \|\mathbb{E}[Ub(\hat{p}(X), a)]\|_2 - \|\hat{\mathbb{E}}_{\mathcal{D}}[Ub(\hat{p}(X), a)]\|_2 \tag{14}$$

$$\leq \sum_a \sup_{b \in B_a^K} \|\mathbb{E}[Ub(\hat{p}(X), a)]\|_2 - \|\hat{\mathbb{E}}_{\mathcal{D}}[Ub(\hat{p}(X), a)]\|_2 \qquad \text{Jensen} \tag{15}$$

$$\leq \sum_a \sup_{b \in B_a^K} \|\mathbb{E}[Ub(\hat{p}(X), a)] - \hat{\mathbb{E}}_{\mathcal{D}}[Ub(\hat{p}(X), a)]\|_2 \qquad \text{Triangle} \tag{16}$$

$$\leq \sum_a \sqrt{C}\mathcal{R}_N(B_a^K) + \sqrt{\frac{2C}{N} \log \frac{2C}{\delta}} \quad (\text{w.p. } 1 - \delta) \qquad \text{Lemma 2} \tag{17}$$

$$\leq \sum_a \sqrt{C} O\left(\sqrt{\frac{CK \log K \log N}{N}}\right) + \sqrt{\frac{2C}{N} \log \frac{2C}{\delta}} \qquad \text{Lemma 3} \tag{18}$$

$$\leq K\sqrt{C} O\left(\sqrt{\frac{CK \log K \log N}{N}}\right) + K\sqrt{\frac{2C}{N} \log \frac{2C}{\delta}} \tag{19}$$

$$= \tilde{O}\left(\frac{K^{3/2}C}{\sqrt{N}}\right) \tag{20}$$

□

**Formal Statement Theorem 2, Part II**

**Theorem 2.2.** *Given any input $\hat{p}$ and tolerance $\epsilon$, Algorithm 1 terminates in $O(K/\epsilon^2)$ iterations. For any $\lambda > 0$, given $O(poly(K, C, \log(1/\delta), \lambda))$ samples, with $1 - \delta$ probability Algorithm 1 outputs a $(\mathcal{L}^K, \epsilon + \lambda)$-decision calibrated prediction function $\hat{p}'$ that satisfies $\mathbb{E}[\|\hat{p}'(X) - Y\|_2^2] \leq \mathbb{E}[\|\hat{p}(X) - Y\|_2^2] + \lambda$.*

*Proof of Theorem 2.2.* We adapt the proof strategy in (Hébert-Johnson et al., 2018). The key idea is to show that a potential function must decrease after each iteration of the algorithm. We choose the potential function as $\hat{\mathbb{E}}[(Y - \hat{p}(X))^2]$. Similar to Appendix A at each iteration $t$ we first lighten

our notation with a shorthand $b_a(X) = b(\hat{p}^{(t-1)}(X), a)$ (at different iteration $t$, $b_a$ denotes different functions), and $b(X)$ is the vector of $(b_1(X), \cdots, b_K(X))$. If the algorithm did not terminate that implies that $b$ satisfies

$$\sum_a \|\hat{\mathbb{E}}[(\hat{p}(X) - Y)b_a(X)]\| \geq \epsilon \tag{21}$$

Denote $\gamma \in \mathbb{R}^{K\times K}$ where $\gamma_a = \hat{\mathbb{E}}[(Y - \hat{p}(X))b(X,a)]/\hat{\mathbb{E}}[b(X,a)]$. The adjustment we make is $\hat{p}'(X) = \pi(\hat{p}(X) + \sum_a \gamma_a b(X,a))$

$$\begin{aligned}
&\hat{\mathbb{E}}[\|Y - \hat{p}(X)\|^2] - \hat{\mathbb{E}}[\|Y - \hat{p}'(X)\|^2] \\
&= \hat{\mathbb{E}}[\|Y - \hat{p}(X)\|^2 - \|Y - \pi(\hat{p}(X) + \sum_a \gamma_a b(X,a))\|^2] \\
&\geq \hat{\mathbb{E}}[\|Y - \hat{p}(X)\|^2 - \|Y - \hat{p}(X) - \sum_a \gamma_a b(X,a)\|^2] && \text{Projection ineq} \\
&= \hat{\mathbb{E}}\left[(2Y - 2\hat{p}(X) - \sum_a \gamma_a b(X,a))^T \left(\sum_a \gamma_a b(X,a)\right)\right] && a^2 - b^2 = (a+b)(a-b) \\
&= 2\sum_a \gamma_a^T \gamma_a \hat{\mathbb{E}}[b(X,a)] - \sum_{a,a'} \gamma_a^T \gamma_{a'} \hat{\mathbb{E}}[b(X,a)b(X,a')] && \text{Definition}\gamma_a \\
&= 2\sum_a \gamma_a^T \gamma_a \hat{\mathbb{E}}[b(X,a)] - \sum_a \gamma_a^T \gamma_a \hat{\mathbb{E}}[b(X,a)b(X,a)] && b(x,a)b(x,a') = 0, \forall a \neq a' \\
&= 2\sum_a \gamma_a^T \gamma_a \hat{\mathbb{E}}[b(X,a)] - \sum_a \gamma_a^T \gamma_a \hat{\mathbb{E}}[b(X,a)] && b(X,a)^2 = b(X,a) \\
&= \sum_a \|\gamma_a\|^2 \hat{\mathbb{E}}[b(X,a)] \geq \frac{1}{K}\left(\sum_a \|\gamma_a\| \hat{\mathbb{E}}[b(X,a)]\right)^2 && \text{Norm inequality} \\
&= \frac{1}{K}\left(\sum_a \left\|\hat{\mathbb{E}}[(Y - \hat{p}(X)b(X,a)]\right\|\right)^2 \geq \frac{\epsilon^2}{K} && \text{Definition } \gamma_a
\end{aligned}$$

Because initially for the original predictor $\hat{p}$ we must have

$$\hat{\mathbb{E}}[\|p^*(X) - \hat{p}(X)\|_2^2] \leq \hat{\mathbb{E}}[\|p^*(X) - \hat{p}(X)\|_1^2] \leq 1$$

the algorithm must converge in $K/\epsilon^2$ iterations and output a predictor $\hat{p}'$ where

$$\sum_a \|\hat{\mathbb{E}}[(\hat{p}'(X) - Y)b_a(X)]\| \leq \epsilon$$

In addition we know that

$$\hat{\mathbb{E}}[\|\hat{p}'(X) - Y\|_2^2] \leq \hat{\mathbb{E}}[\|\hat{p}(X) - Y\|_2^2]$$

Now that we have proven the theorem for empirical averages (i.e. all expectations are $\hat{\mathbb{E}}$), we can convert this proof to use true expectations (i.e. all expectations are $\mathbb{E}$) by observing that all expectations involved in the proof satisfy $\mathbb{E}[\cdot] \in \hat{\mathbb{E}}[\cdot] \pm \kappa$ for any $\kappa > 0$ and sample size that is polynomial in $\kappa$. □

*Proof of Theorem 2.2'.* Observe that the matrix $D$ defined in Algorithm 2 is a symmetric, positive semi-definite and non-negative matrix such that $\sum_{a,a'} D_{aa'} = 1$. To show that the algorithm converges we first need two Lemmas on the properties of such matrices. For a positive semi-definite (PSD) symmetric matrix, let $\lambda_1$ denote the largest eigenvalue, and $\lambda_n$ denote the smallest eigenvalue (which are always real numbers). The first Lemma is a simple consequence of the Perron-Frobenius theorem,

**Lemma 4.** *Let $D$ be any symmetric PSD non-negative matrix such that $\sum_{a,a'} D_{aa'} = 1$, then $\lambda_1(D) \leq 1$, so $\lambda_n(D^{-1}) \geq 1$.*

**Lemma 5** ((Fang et al., 1994) Theorem 1)**.** *Let $D$ be a symmtric PSD matrix, then for any matrix $B$ (that has the appropriate shape to multiply with $D$)*

$$\lambda_n(D)\text{trace}(B) \leq \text{trace}(BD) \leq \lambda_1(D)\text{trace}(B)$$

Now we can proveed to prove our main result. We have to show that the L2 error $\hat{\mathbb{E}}[(Y-\hat{p}^{(t-1)}(X))^2]$ must decrease at iteration $t$ if we still have

$$\epsilon^2/K \leq \sum_a \|\hat{\mathbb{E}}[(Y-\hat{p}^{(t-1)}(X))b_a^{(t)}(X)]\|^2 := \text{trace}(RR^T)$$

We can compute the reduction in L2 error after the adjustment

$$\begin{aligned}
&\hat{\mathbb{E}}[(Y-\hat{p}^{(t-1)}(X))^2] - \hat{\mathbb{E}}[(Y-\hat{p}^{(t)}(X))^2] \\
&= \hat{\mathbb{E}}\left[(2(Y-\hat{p}^{(t-1)}(X)) - R^TD^{-1}b^{(t)}(X))^TR^TD^{-1}b^{(t)}(X)\right] && \text{Definition} \\
&= 2\hat{\mathbb{E}}\left[(Y-\hat{p}^{(t-1)}(X))^TR^TD^{-1}b^{(t)}(X)\right] - \hat{\mathbb{E}}\left[b^{(t)}(X)^TD^{-T}RR^TD^{-1}b^{(t)}(X)\right] \\
&= 2\text{trace}\left(\hat{\mathbb{E}}\left[b^{(t)}(X)(Y-\hat{p}^{(t-1)}(X))^TR^TD^{-1}\right]\right) \\
&\quad - \text{trace}\left(\hat{\mathbb{E}}\left[b^{(t)}(X)b^{(t)}(X)^TD^{-T}RR^TD^{-1}\right]\right) && \text{Cyclic property} \\
&= 2\text{trace}\left(RR^TD^{-1}\right) - \text{trace}(RR^TD^{-1}) = \text{trace}(RR^TD^{-1}) && \text{Definition} \\
&\geq \text{trace}(RR^T) \geq \epsilon^2/K && \text{Lemma 5 and 4}
\end{aligned}$$

Therefore, the algorithm cannot run for more than $O(K/\epsilon^2)$ iterations. Suppose the algorithm terminates we must have

$$\begin{aligned}
\sup_{b\in B^K}\sum_a \|\hat{\mathbb{E}}[(Y-\hat{p}^{(t-1)}(X))b_a^{(t)}(X)]\| &\leq \sup_{b\in \bar{B}^K}\sum_a \|\hat{\mathbb{E}}[(Y-\hat{p}^{(t-1)}(X))b_a^{(t)}(X)]\| \\
&\leq \sup_{b\in \bar{B}^K}\sqrt{K}\sqrt{\sum_a \|\hat{\mathbb{E}}[(Y-\hat{p}^{(t-1)}(X))b_a^{(t)}(X)]\|^2} \\
&\leq \sqrt{K}\epsilon/\sqrt{K} = \epsilon
\end{aligned}$$

So by Proposition 3 we can conclude that the algorithm must output a $(\mathcal{L}^K, \epsilon)$-decision calibrated prediction function. □

### C.5 Proof of Remaining Lemmas

*Proof of Lemma 1.* By the orthogonal property of the condition expectation, for any event $A$ in the sigma algebra induced by $V$, we have

$$\mathbb{E}[(U - \mathbb{E}[U \mid V])\mathbb{I}_A] = 0$$

This includes the event $V > c$

$$\mathbb{E}[(U - \mathbb{E}[U \mid V])\mathbb{I}(V > c)] = 0$$

In other words,

$$\mathbb{E}[U\mathbb{I}(V > c)] = \mathbb{E}[\mathbb{E}[U \mid V]\mathbb{I}(V > c)]$$

□

*Proof of Lemma 2.* First observe that by the norm inequality $\|z\|_\alpha \leq C^{1/\alpha}\|z\|_\infty$ we have

$$\|\hat{\mathbb{E}}_{\mathcal{D}}[Ub(V)] - \mathbb{E}[Ub(V)]\|_2 \leq \sqrt{C}\|\hat{\mathbb{E}}[Ub(V)] - \mathbb{E}[Ub(V)]\|_\infty \tag{22}$$

Denote the $c$-th dimension of $U$ by $U^c$; we now provide bounds for $|\hat{\mathbb{E}}[U^cb(V)] - \mathbb{E}[U^cb(V)]|$ by standard Radamacher complexity arguments. Define a set of ghost samples $\bar{\mathcal{D}} = \{(\bar{U}_1, \bar{V}_1), \cdots (\bar{U}_N, \bar{V}_N)\}$ and Radamacher variables $\sigma_n \in \{-1, 1\}$

$$\mathbb{E}_{\mathcal{D}}\left[\sup_b |\hat{\mathbb{E}}_{\mathcal{D}}[U^cb(V)] - \mathbb{E}[U^cb(V)]|\right] \tag{23}$$

$$= \mathbb{E}_{\mathcal{D}}\left[\sup_b \left|\hat{\mathbb{E}}_{\mathcal{D}}[U^c b(V)] - \mathbb{E}_{\bar{\mathcal{D}}}\left[\hat{\mathbb{E}}_{\bar{\mathcal{D}}}[U^c b(V)]\right]\right|\right] \quad \text{Tower} \tag{24}$$

$$= \mathbb{E}_{\mathcal{D}}\left[\sup_b \left|\mathbb{E}_{\bar{\mathcal{D}}}[\hat{\mathbb{E}}_{\mathcal{D}}[U^c b(V)] - \hat{\mathbb{E}}_{\bar{\mathcal{D}}}[U^c b(V)]]\right|\right] \quad \text{Linearity} \tag{25}$$

$$\leq \mathbb{E}_{\mathcal{D}}\left[\sup_b \mathbb{E}_{\bar{\mathcal{D}}}\left[\left|\hat{\mathbb{E}}_{\mathcal{D}}[U^c b(V)] - \hat{\mathbb{E}}_{\bar{\mathcal{D}}}[U^c b(V)]\right|\right]\right] \quad \text{Jensen} \tag{26}$$

$$\leq \mathbb{E}_{\mathcal{D},\bar{\mathcal{D}}}\left[\sup_b \left|\hat{\mathbb{E}}_{\mathcal{D}}[U^c b(V)] - \hat{\mathbb{E}}_{\bar{\mathcal{D}}}[U^c b(V)]\right|\right] \quad \text{Jensen} \tag{27}$$

$$\leq \mathbb{E}_{\sigma,\mathcal{D},\bar{\mathcal{D}}}\left[\sup_b \left|\frac{1}{N}\sum_i \sigma_i U_i^c b(V_i) - \frac{1}{N}\sum_i \sigma_i \bar{U}_i^c b(\bar{V}_i)\right|\right] \quad \text{Radamacher} \tag{28}$$

$$\leq \mathbb{E}_{\sigma,\mathcal{D},\bar{\mathcal{D}}}\left[\sup_b \left|\frac{1}{N}\sum_i \sigma_i U_i^c b(V_i)\right| + \sup_b \left|\frac{1}{N}\sum_i \sigma_i \bar{U}_i^c b(\bar{V}_i)\right|\right] \quad \text{Jensen} \tag{29}$$

$$= 2\mathbb{E}_{\sigma,\mathcal{D}}\left[\sup_b \left|\frac{1}{N}\sum_i \sigma_i U_i^c b(V_i)\right|\right] \tag{30}$$

Suppose we know the Radamacher complexity of the function family $b$

$$\mathcal{R}_N(B) := \mathbb{E}\left[\sup_b \frac{1}{N}\sum_i \sigma_i b(V_i)\right] \tag{31}$$

Then by the contraction inequality, and observe that $U_i^c \in [-1, 1]$ so multiplication by $U_i^c$ is a 1-Lipschitz map, we can conclude for any $c \in [C]$

$$\mathcal{R}_N(B) \geq \mathbb{E}\left[\sup_b \frac{1}{N}\sum_i \sigma_i U_{ic} b(V_i)\right] \tag{32}$$

Finally observe that the map $\mathcal{D} \to \frac{1}{N}\sum_i \sigma_i U_{ic} b(V_i)$ has $2/N$ bounded difference, so by Mcdiamid inequality for any $\epsilon > 0$

$$\Pr\left[\sup_b \left|\frac{1}{N}\sum_i \sigma_i U_i^c b(V_i)\right| \geq \mathcal{R}_N(B) + \epsilon\right] \leq 2e^{-N\epsilon^2/2} \tag{33}$$

By union bound we have

$$\Pr\left[\max_c \sup_b \left|\frac{1}{N}\sum_i \sigma_i U_i^c b(V_i)\right| \geq \mathcal{R}_N(B) + \epsilon\right] \leq 2Ce^{-N\epsilon^2/2} \tag{34}$$

We can combine this with Eq.(22) to conclude

$$\Pr\left[\sup_b \|\hat{\mathbb{E}}[Ub(V)] - \mathbb{E}[Ub(V)]\|_2 \geq \sqrt{C}\mathcal{R}_N(B) + \sqrt{C}\epsilon\right]$$
$$\leq \Pr\left[\max_c \sup_b \hat{\mathbb{E}}[U^c b(V)] - \mathbb{E}[U^c b(V)]\|_2 \geq \mathcal{R}_N(B) + \epsilon\right] \leq 2Ce^{-N\epsilon^2/2}$$

Rearranging we get $\forall \delta > 0$

$$\Pr\left[\sup_b \|\hat{\mathbb{E}}[Ub(V)] - \mathbb{E}[Ub(V)]\|_2 \geq \sqrt{C}\mathcal{R}_N(B) + \sqrt{\frac{2C}{N}\log\frac{2C}{\delta}}\right] \leq \delta$$

□

*Proof of Lemma 3.* For $B_a^K$ we use the VC dimension approach. Because $\forall b \in B_a^K$ the set $\{z \in \Delta^C, b(z) = 1\}$ is the intersection of $K$ many $C$-dimensional half plances, its VC dimension $\mathrm{VC}(B_a^K) \leq (C+1)2K\log_2(3K)$ (Mohri et al., 2018) (Q3.23). By Sauer's Lemma we have

$$\mathcal{R}_N(B_a^K) \leq \sqrt{\frac{2\mathrm{VC}(B_a^K)\log(eN/\mathrm{VC}(B_a^K))}{N}} = O\left(\sqrt{\frac{2CK\log K\log N}{N}}\right)$$

□