Reliable Decisions with Threshold Calibration

Roshni Sahoo
Stanford University
rsahoo@stanford.edu

Shengjia Zhao
Stanford University
sjzhao@stanford.edu

Alyssa Chen
UTSW Medical Center
alyssa.chen@utsw.edu

Stefano Ermon
Stanford University
ermon@stanford.edu

Abstract

Decision makers rely on probabilistic forecasts to predict the loss of different decision rules before deployment. When the forecasted probabilities match the true frequencies, predicted losses will be accurate. Although perfect forecasts are typically impossible, probabilities can be calibrated to match the true frequencies on average. However, we find that this *average* notion of calibration, which is typically used in practice, does not necessarily guarantee accurate decision loss prediction. Specifically in the regression setting, the loss of threshold decisions, which are decisions based on whether the forecasted outcome falls above or below a cutoff, might not be predicted accurately. We propose a stronger notion of calibration called threshold calibration, which is exactly the condition required to ensure that decision loss is predicted accurately for threshold decisions. We provide an efficient algorithm which takes an uncalibrated forecaster as input and provably outputs a threshold-calibrated forecaster. Our procedure allows downstream decision makers to confidently estimate the loss of any threshold decision under any threshold loss function. Empirically, threshold calibration improves decision loss prediction without compromising on the quality of the decisions in two real-world settings: hospital scheduling decisions and resource allocation decisions.

1 Introduction

Decision makers need to understand the consequences of their decisions prior to making them. When decisions are based on predictions from a machine learning model, the decision loss – the loss incurred under a decision rule based on the predictions – summarizes the consequences of these decisions. As an example, suppose a machine learning practitioner develops a model to predict patient length-of-stay in the hospital [17, 3]. A hospital decides whether they have capacity to admit new patients based on the model's predictions of current patients' length-of-stay (e.g. for each current patient who is predicted to have a length-of-stay that is less than k days, the hospital schedules a new patient). Incorrect decisions due to the model's predictions cause the hospital to accrue costs from under-utilizing resources or overbooking procedures. The decision loss is an aggregation of the costs incurred from incorrect decisions. To determine whether a decision rule is safe to use, the hospital would like to have an accurate estimate of the decision loss under different choices of k and different costs associated with errors. This type of decision-making scenario occurs in many high-stakes settings such as designing interventions for adverse weather events [33, 9] and resource allocation decisions using economic estimates [15, 32].

Probabilistic predictions (probabilistic forecasts) can be used to estimate decision loss prior to deployment. In this work, we consider the regression setup, where a forecast is represented by a cumulative distribution function over the possible outcomes. If the forecasted probabilities of

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

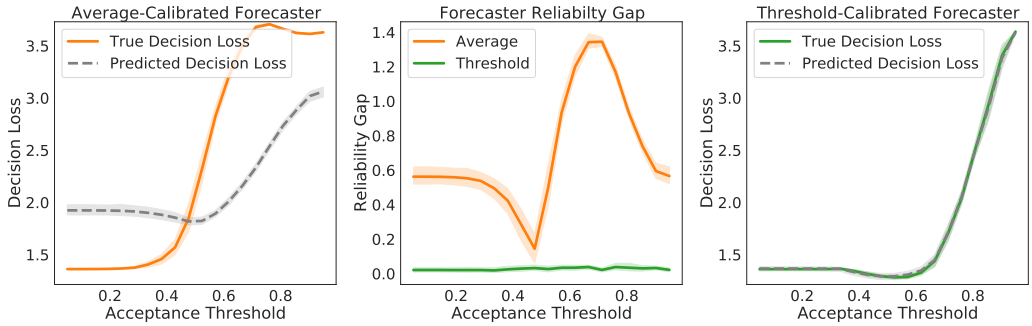

Figure 1: We evaluate average-calibrated and threshold-calibrated patient length-of-stay forecasters across a range of threshold decision rules. **Left**: The average-calibrated forecaster underestimates the true decision loss for some decision rules and overestimates it on others, resulting in a nonzero reliability gap. **Middle**: The reliability gap is minimized under the threshold-calibrated forecaster but not under the average-calibrated forecaster. **Right**: The threshold-calibrated forecaster accurately predicts the true decision loss across a range of decision rules.

incorrect decisions match the true frequencies of these events, then the average decision loss can be accurately predicted from the forecasts. However, forecasted probabilities of incorrect decisions do not typically match the true ones, yielding inaccurate decision loss predictions. We refer to the absolute difference between the average loss predicted by forecaster and the true average loss as the **reliability gap**.

Many previous works in calibration and uncertainty quantification are motivated by the assumption that calibrated uncertainty estimates will yield safer or more reliable downstream decisions [31, 2, 22, 24, 25]. However, we find that the standard notion of calibration, average calibration, does not guarantee zero reliability gap for even a simple class of decision rules: threshold decision rules (left, Figure 1). In a threshold decision, a decision maker takes one of two possible actions depending on whether an outcome falls above or below a cutoff (e.g. the hospital schedules a new patient if a current patient's length-of-stay is less than 3 days, otherwise the hospital does not schedule a new patient). Stronger calibration properties, such as distribution calibration [31], are theoretically guaranteed to yield zero reliability gap but are difficult to achieve in practice. In particular, flexible distribution families can better approximate the true distribution than simple ones and yield lower decision loss, but applying distribution calibration to such forecasters can increase the decision loss and enlarge the reliability gap compared to the uncalibrated forecaster. Thus, existing calibration definitions are either insufficient or impractical for minimizing the reliability gap under threshold decision rules.

To address these shortcomings, we propose a new notion of calibration called **threshold calibration**. Threshold calibration strikes a balance between average and distribution calibration; it is exactly the condition required to guarantee zero reliability gap under threshold decisions and is practical to enforce (Figure 1, Right). First, we establish that threshold calibration is the necessary and sufficient condition to guarantee zero reliability gap for any threshold decision under any threshold loss. Second, we design an *efficient* algorithm that takes an uncalibrated forecaster as input and provably outputs a threshold-calibrated forecaster. Third, we show that empirically, threshold calibration is a *practical* solution; in two real-world settings and a suite of benchmark regression tasks, we find that threshold calibration minimizes the reliability gap across decision makers with different threshold loss functions while achieving similar or improved decision loss compared to the baselines.

2 Preliminaries

2.1 Notation and Forecasting Setup

We consider the regression setup with a feature space $\mathcal{X}$ and a label space $\mathcal{Y} \subset \mathbb{R}$. The input is a random variable $X \in \mathcal{X}$ and the label is a random variable $Y \in \mathcal{Y}$. We use capital letters to denote random variables X, Y and lower case letters to denote their values x, y.

Let $\mathcal{F}(\mathcal{Y})$ be the space of cumulative distribution functions (CDFs) over $\mathcal{Y}$. A forecaster $h : \mathcal{X} \to \mathcal{F}(\mathcal{Y})$ is a function that maps an input from the feature space to a CDF on $\mathcal{Y}$. In other

words, given a fixed input $x \in \mathcal{X}$, the forecaster outputs the predicted CDF $h[x] \in \mathcal{F}(\mathcal{Y})$. Ideally, the forecaster aims to predict the CDF of Y given X.

To further clarify the notation, for a fixed input-label pair $(x, y) \in \mathcal{X} \times \mathcal{Y}$, $h[x]$ is a CDF over the predicted label values and $h[x](y) \in [0, 1]$ is the value of the CDF $h[x]$ at the point y. We note that $h[X]$ is a random variable that takes values in $\mathcal{F}(\mathcal{Y})$ and $h[X](Y)$ is a random variable that takes values in $[0, 1]$.

Let $h^*[X]$ be the true conditional CDF of Y given X. We use $\sim$ to denote the distribution of a random variable. We have that $Y \sim h^*[X]$. We introduce a new random variable $\tilde{Y}$ to represent a label distributed according to the $h[X]$, the forecasted conditional distribution, so $\tilde{Y} \sim h[X]$.

2.2 Decision-Making

Let $\mathcal{A}$ be a countable action space. A decision rule $\delta : \mathcal{X} \to \mathcal{A}$ is any map from an input x (e.g. a current patient's attributes) to an action a (e.g. admit a new patient). We assume that a decision maker has a loss function $\ell : \mathcal{X} \times \mathcal{Y} \times \mathcal{A} \to \mathbb{R}$, describing the loss incurred when choosing an action a on an input-label pair (x, y). Because the labels y are unobserved, the decision maker often wants to minimize their expected loss assuming that the labels are distributed according to the forecasted distribution. As a result, they use the Bayes decision rule with respect to h.

Definition 1 (Bayes Decision Rule)**.** *Given a space of decision rules Δ, the Bayes decision rule with respect to the forecaster h is the decision rule in Δ that minimizes the expected loss under the forecasted distribution*

$$\delta_h^* = \arg\inf_{\delta \in \Delta} \mathbb{E}_X \mathbb{E}_{\tilde{Y} \sim h[X]}[\ell(X, \tilde{Y}, \delta(X))]]$$

2.3 Threshold Decisions

We focus on the setting where the decision maker aims to minimize a threshold loss function. The action space $\mathcal{A}$ consists of two actions so $\mathcal{A} = \{0, 1\}$. A threshold loss function ℓ is defined as follows

$$\ell(x, y, a) = \sum_{i \in \{0,1\}} c_{1,i} \mathbb{I}(y \leq y_0, a = i) + \sum_{i \in \{0,1\}} c_{0,i} \mathbb{I}(y > y_0, a = i),$$

where $c_{i,j} \in \mathbb{R}$. The $c_{i,j}$'s denote *decision costs*, costs associated with different outcome-action pairs, and y_0 is a *decision threshold*. Let $\mathcal{L}$ be the space of threshold loss functions, which are all losses of this form with any $c_{i,j} \in \mathbb{R}$ and $y_0 \in \mathbb{R}$.

Given a threshold loss function ℓ, the decision maker can use the Bayes decision rule δ_h^* in Definition 1 to select which action to take. We show that the resulting decision rules always take the form of

$$\delta_h^*(x) = \mathbb{I}(h[x](y_0) \geq \alpha) \text{ or } \delta_h^*(x) = \mathbb{I}(h[x](y_0) \leq \alpha)$$

for some parameters $\alpha \in [0, 1]$ and $y_0 \in \mathcal{Y}$ that depends on the loss function (proved in Appendix B). We call such decision rules **threshold decision rules** because intuitively, they choose the action based on whether $h[x](y_0)$ is greater (or less) than a threshold α. We denote the space of such decision rules as Δ_h. Since the decision maker's loss function is a threshold loss function, the decision maker can restrict the space of decision rules they consider to threshold decision rules on the forecasted CDFs.

3 Reliable Decision-Making with Threshold Calibration

3.1 Problem Setup

Forecasts are often produced by one group, such as machine learning practitioners or scientists, and consumed by another, such as policy makers or private agents [14]. Motivated by this paradigm, we model these two entities separately:

1. A forecaster h takes inputs $x \in \mathcal{X}$ and produces CDFs $h[x]$ over the possible outcomes in $\mathcal{Y}$. The provider of h does not know the specific downstream tasks for which h is used.

2. A decision maker has a dataset of unlabeled inputs $\mathcal{D} = \{x_i\}_{i=1}^n$, binary action space $\mathcal{A} = \{0, 1\}$, and a threshold loss function $\ell \in \mathcal{L}$ of interest. The decision maker must take an action $a_i \in \mathcal{A}$ for each unlabeled input x_i. The decision maker uses the forecaster h to select $\{a_i\}_{i=1}^n$ because (1) the decision maker does not have enough labeled data to build their own model locally or (2) building the model requires a domain expert.

Multiple decision makers may rely on the same forecaster but have different loss functions. Further, a decision maker's loss function can change if their decision costs or decision threshold change. In this setting, we identify the conditions on h that the provider can enforce to ensure reliable decision-making under threshold decisions.

3.2 Reliability Gap

Decision makers often need to accurately estimate the average decision loss incurred under a decision rule prior to deployment. To quantify the accuracy of these decision loss predictions, we define the reliability gap.

Definition 2 (Reliability Gap)**.** *Given a forecaster h, we define the the reliability gap $\gamma(\delta, \ell)$ of a particular decision rule δ under a loss function ℓ as*

$$\gamma(\delta, \ell) = |\mathbb{E}_X \mathbb{E}_{\tilde{Y} \sim h[X]}[\ell(X, \tilde{Y}, \delta(X))] - \mathbb{E}_X \mathbb{E}_{Y \sim h^*[X]}[\ell(X, Y, \delta(X))]|.$$

The first term in the equation is the average decision loss predicted by the forecaster. Under the forecasted distribution, the labels $\tilde{Y}$ are distributed according to $h[X]$. As a result, the first term does not depend on the true labels and can be computed by the decision maker using the unlabeled data prior to deployment. The second term is the true average decision loss. Under the true conditional distribution, the labels Y are distributed according to $h^*[X]$. So, the second term can be thought of as the loss that is incurred at test-time. One caveat is that the reliability gap quantifies the reliability of *average* decision loss prediction and obtaining zero reliability gap does not imply any instance-based guarantees for individual decisions.

When the forecaster perfectly matches the true distribution (i.e. $h = h^*$), we have $\gamma(\delta, \ell) = 0$ for any decision rule δ and any loss function ℓ. However, in practice, we cannot assume that the forecaster predicts the true distribution. In addition, we would like the forecaster to be applicable for different downstream decision makers. As a result, we study the necessary and sufficient conditions on the forecaster that guarantee zero reliability gap for any threshold decision on the forecasted CDFs and any threshold loss function.

3.3 Threshold Calibration

We define the property of threshold calibration and show that it is necessary and sufficient to ensure zero reliability gap under any threshold decision on the forecasted CDFs and any threshold loss function. The lemma and theorem in this section are proven in Appendix B.

We define the property of threshold calibration below.

Definition 3 (Threshold Calibration)**.** *A forecaster h satisfies threshold calibration if*

$$\Pr[h[X](Y) \leq c \mid h[X](y_0) \leq \alpha] = c \quad \forall y_0 \in \mathcal{Y}, \alpha \in [0, 1], \forall c \in [0, 1]. \tag{1}$$

A threshold-calibrated forecaster is average-calibrated on subsets of the predicted CDFs that satisfy $h[X](y_0) \leq \alpha$.We make the following observation about conditioning on the complementary predicted CDFs.

Lemma 1. *Given a forecaster h that satisfies Definition 3, then we have that $\forall y_0 \in \mathcal{Y}, \alpha \in [0, 1], \forall c \in [0, 1], \Pr[h[X](Y) \leq c \mid h[X](y_0) > \alpha] = c$.*

In a threshold decision task, a decision maker will take action a given inputs with predicted CDFs satisfying $h[X](y_0) \leq \alpha$ (and take a complementary action given inputs with predicted CDFs satisfying $h[X](y_0) > \alpha$). Intuitively, threshold calibration ensures that the forecaster satisfies average calibration on the subsets of predicted CDFs where the decision maker chooses $a = 0$ and $a = 1$.

Threshold calibration is a specific type of group calibration [28], where calibration across the collection of groups $\mathcal{G} = \{(X, Y) \in \mathcal{X} \times \mathcal{Y} \mid h[X](y_0) \leq \alpha\}_{y_0 \in \mathcal{Y}, \alpha \in [0,1]}$ is desired. Since threshold calibration requires achieving calibration on intersecting groups, it is also related to the notion of multicalibration [18]. In Section 4, we give an efficient algorithm for achieving threshold calibration that is inspired by previous work on multicalibration.

Using Definition 3 and Lemma 1, we define the threshold calibration error (TCE) to measure deviation from threshold calibration at a threshold $y_0 \in \mathcal{Y}$ and quantile $\alpha \in [0, 1]$.

Definition 4 (Threshold Calibration Error)**.**

$$\begin{aligned} \textit{TCE}(h, y_0, \alpha) = &\int_0^1 |\Pr[h[X](Y) \leq c \mid h[X](y_0) \leq \alpha] - c| \quad dc \\ &+ \int_0^1 |\Pr[h[X](Y) \leq c \mid h[X](y_0) > \alpha] - c| \quad dc. \end{aligned}$$

Threshold calibration is a desirable property due to its connection to achieving zero reliability gap.

Theorem 1. *Let $\mathcal{L}$ be the space of threshold loss functions. Given a forecaster h, let Δ_h be the space of threshold decision rules on the forecasted CDFs of h. A forecaster h satisfies threshold calibration if and only if $\gamma(\delta, \ell) = 0 \quad \forall \delta \in \Delta_h, \forall \ell \in \mathcal{L}$.*

We obtain this result by observing that the expected decision loss under the true distribution can be decomposed into two terms. The first term corresponds to the cost incurred from "false positive" errors and the second term corresponds to the cost incurred from "false negative" errors. Under threshold calibration, the forecaster's predicted error rates match the true error rates. Since the decision loss (with any choice of costs) is a linear combination of these error rates, the expected decision loss predicted by the forecaster matches the expected decision loss under the true distribution. Thus, under a threshold-calibrated forecaster, we achieve zero reliability gap under any threshold decision on the forecasted CDFs and any threshold loss function.

3.4 Comparison to Existing Calibration Definitions

We compare threshold calibration to other methods for calibrating probabilistic forecasts. Average calibration is the standard definition of calibration for regression [23, 12].

Definition 5 (Average Calibration)**.** *A forecaster h satisfies average calibration if*

$$\Pr[h[X](Y) \leq c] = c \quad \forall c \in [0, 1].$$

In other words, a forecaster is average-calibrated if the true label Y is below the c-th quantile of the forecasted CDF $h[x]$ exactly c percent of the time.

In contrast, distribution calibration is a much stronger definition of calibration [31]. Intuitively, distribution calibration requires a forecaster to be calibrated for every distribution in the forecaster's model family.

Definition 6 (Distribution Calibration)**.** *A forecaster h satisfies distribution calibration if*

$$\Pr[h[X](Y) \leq c \mid h[X] = g] = c \quad \forall g \in \mathcal{F}(\mathcal{Y}),$$

where $\mathcal{F}$ is space of CDFs corresponding to the forecaster's model family.

We outline the relationship between average, threshold, and distribution calibration in the following proposition.

Proposition 1. *If a forecaster satisfies distribution calibration, then it satisfies threshold calibration. If a forecaster satisfies threshold calibration, then it satisfies average calibration.*

We note that the converses of the statements in Proposition 1 are not necessarily true. A threshold-calibrated forecaster does not necessarily satisfy distribution calibration. An average-calibrated forecaster does not necessarily satisfy threshold calibration or distribution calibration (see Appendix C). This implies that an average-calibrated forecaster does not satisfy the necessary condition of Theorem 1, meaning that the reliability gap under threshold decisions may not be zero. So, decision

makers who rely on a forecaster that only satisfies average calibration (but not threshold calibration) are not guaranteed to accurately estimate their decision loss under threshold decisions.

From Proposition 1, we have that a distribution-calibrated forecaster satisfies the necessary condition of Theorem 1. However, distribution calibration can be challenging to achieve in practice because the same CDF is rarely predicted more than one time on the training samples, making it difficult to guarantee calibration without compromising the *sharpness* of the forecasts. Sharpness corresponds to the width of the prediction intervals generated from the forecasts, and sharp forecasts yield short prediction intervals. Although distribution calibration is theoretically guaranteed to yield zero reliability gap, we observe that achieving distribution calibration is challenging when the model family is complex (Section 5).

Finally, we emphasize the threshold calibration is exactly the condition needed to guarantee the reliability gap is zero in Theorem 1.

4 Achieving Threshold Calibration

We design a recalibration algorithm that takes an uncalibrated forecaster as input and provably outputs a threshold-calibrated forecaster. Our algorithm is an iterative procedure that terminates when the maximum TCE is less than a user specified threshold ϵ. Our key result is that the algorithm must terminate after $O(1/\epsilon^2)$ iterations.

Pseudo-code for the algorithm is shown in Algorithm 1. Intuitively, at each iteration of the algorithm, we find the y_0^t and α^t where the TCE in Definition 4 is maximized. This partitions the input $\mathcal{X}$ into two parts: those where $h[x](y_0^t) \leq \alpha^t$ and those where $h[x](y_0^t) > \alpha^t$. For each partition, we use a standard recalibration algorithm (Isotonic regression [23]) to achieve average calibration. Intuitively, after the recalibration step, the forecaster should satisfy average calibration for each partition, and hence the TCE in Definition 4 must be (close to) 0 for y_0^t and α^t. We repeat this procedure until the TCE is less than ϵ for every possible y_0 and α.

Algorithm 1: Threshold Recalibration

1 **Input**: Forecaster $h : \mathcal{X} \to \mathcal{F}(\mathcal{Y})$, maximum error $\epsilon > 0$
2 **Output**: A threshold-calibrated forecaster
3 Set $h^0 \leftarrow h$
4 **for** $t = 1, 2, \cdots$ *until maximum threshold calibration error* $\sup_{y_0,\alpha} \mathit{TCE}(h^{t-1}, y_0, \alpha) \leq \epsilon$ **do**
5 Find the y_0 and α that maximize threshold calibration error.
$y_0^t, \alpha^t \leftarrow \arg\sup_{(y_0,\alpha)\in\mathcal{Y}\times[0,1]} \text{TCE}(h^{t-1}, y_0, \alpha)$
6 Partition input features $\mathcal{X}$ into $\mathcal{X}_0 \leftarrow \{x \in \mathcal{X} \mid h^{t-1}[x][y_0^t] \leq \alpha^t\}$ and $\mathcal{X}_1 = \mathcal{X} \setminus \mathcal{X}_0$.
7 Use Isotonic regression to learn recalibration maps $\phi_0^t, \phi_1^t : \mathcal{F}(\mathcal{Y}) \to \mathcal{F}(\mathcal{Y})$ on $\mathcal{X}_0$ and $\mathcal{X}_1$ respectively.
8 Apply the recalibration map to obtain new prediction functions.
$h^t[x] \leftarrow \begin{cases} \phi_0^t(h^{t-1}[x]) & \text{if } x \in \mathcal{X}_0 \\ \phi_1^t(h^{t-1}[x]) & \text{otherwise} \end{cases}$
9 **end**
10 **return** h^T where T is the final iteration count.

The following theorem shows that our iterative threshold calibration procedure converges in a small number of iterations. The intuition of the proof is that after each iteration, the L2 distance between the prediction functions h and the true CDF h^* must decrease by at least ϵ^2. Therefore, the algorithm must terminate before the L2 distance decreases below 0 (which is impossible). A full proof is provided in Appendix B.

Theorem 2. *Algorithm 1 converges after at most $O(1/\epsilon^2)$ iterations and outputs a forecaster with threshold calibration error at most ϵ.*

For simplicity, we do not consider finite sample approximation of the TCE in line 5 of Algorithm 1. Line 5 can be interpreted in two ways: line 5 estimates the TCE on the true distribution (which we can only do with infinite samples), or on the empirical distribution (i.e. the uniform distribution on

the recalibration data). Under the former interpretation, Theorem 2 holds assuming that line 5 can estimate the true TCE (which is the ideal scenario with infinite data). Under the latter interpretation, Theorem 2 holds for the empirical distribution, i.e. it guarantees that Algorithm 1 will output a forecaster with threshold calibration error at most ϵ *on the empirical distribution* rather than the true distribution. We will instead use experiments to show that Algorithm 1 can generalize to the true distribution. Note that under both interpretations, Algorithm 1 will converge after at most $O(1/\epsilon^2)$ iterations. For completeness, we describe the finite sample version of the algorithm in Appendix A.

5 Experiments

In the following experiments, we demonstrate that threshold calibration can minimize the reliability gap (1) across a range of decision costs, (2) across a range of decision thresholds, and (3) in simple and complex model families. Across all datasets and forecaster model families that we consider, we find that threshold calibration outperforms the baselines in reducing the size of the reliability gap while attaining similar or improved decision loss compared to the baselines.

5.1 Datasets

We consider datasets that relate to real-world decision-making tasks and standard benchmarks. In the main paper, we show results on the UCI Protein and the MIMIC-III datasets. All remaining results can be found in Appendix A.

MIMIC-III. Patient length-of-stay predictions are used for hospital scheduling and resource management [17]. We consider a patient length-of-stay forecaster trained on patient admission laboratory values from the MIMIC-III dataset [20]. In our decision task, the hospital decides to schedule a new patient for an elective procedure if a current patient is predicted to have a short length of stay.

Demographic and Health Survey (DHS). Local wealth measurements are used to inform resource allocation decisions. We use the DHS data from Sheehan et al. [30] to predict asset wealth from satellite images as done in Yeh et al. [32] and Sheehan et al. [30]. Our experimental setup is motivated by the decision task defined in Yeh et al. [32], where aid is allocated to regions where the predicted asset wealth falls below a particular threshold.

UCI Regression Datasets. We evaluate on a suite of UCI regression datasets (Naval, Protein, Energy, Crime) [11]. They are common benchmarks in the uncertainty quantification literature [31, 2, 8, 23].

5.2 Experimental Setup and Baselines

Experimental Setup. We consider a forecaster that outputs Gaussian distributions and a forecaster that outputs Gaussian-Laplace mixture distributions. We use a train/validation/test split. The uncalibrated forecaster is a neural network trained on the training set with the validation set used for early stopping. For large datasets (Protein, Energy, Naval, MIMIC-III), the recalibration transform is trained on the validation set. For small datasets (Crime, DHS), the recalibration transform is trained on the training and validation set. On the test set, we evaluate our method and the baselines using decision-making metrics (Section 5.3). Calibration metrics are also measured and results are provided in Appendix A.

Baselines. We compare the uncalibrated forecaster to the forecaster after enforcing average, threshold, or distribution calibration through a posthoc recalibration procedure. Methods for achieving these properties are described in Appendix A.

5.3 Decision-Making Metrics

We simulate decision makers enumerated $i = 1, 2 \dots M$ who use a probabilistic forecaster h for their threshold decision tasks. We assume that there is no cost associated with true positives or true negatives, and the total cost of a false positive plus a false negative is equal to 10 for all decision makers. As a result, decision maker i's task is determined by a decision threshold y_0^i and decision cost ratio c_i. Each decision maker has a loss function $\ell^i(x, y, a) = 10c_i\mathbb{I}(a = 1, y \geq y_0^i) + 10(1 - c_i)\mathbb{I}(a = 0, y < y_0^i)$ and a decision rule

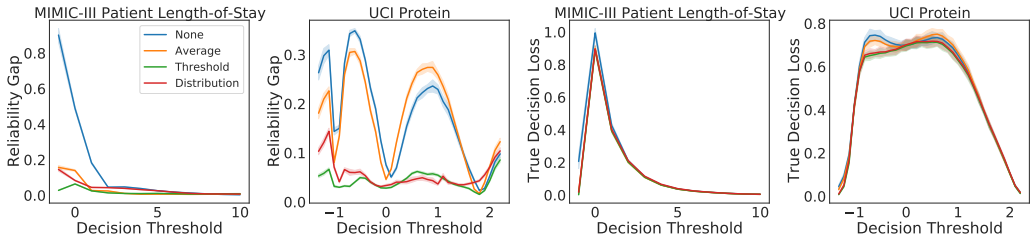

Figure 2: Under the Gaussian forecaster and across different decision thresholds, threshold calibration reduces the reliability gap on both datasets while average calibration does not reduce the reliability gap on the Protein dataset (**Left, Middle Left**), and all calibration methods yield improved or comparable decision loss compared to the uncalibrated forecaster (**Middle Right, Right**). Error bars represent 95% confidence intervals and are generated over 6 random trials.

$\delta^i_{h,\alpha}(x) = \mathbf{1}(h[x](y^i_0) \geq \alpha)$. We consider decision makers with $(y^i_0, c_i) \in \mathcal{Y}_0 \times \mathcal{C}$ where $\mathcal{Y}_0$ and $\mathcal{C}$ each consist of 50 uniformly spaced points that span the label space and [0.05, 0.95], respectively.

For each decision maker i, we compute the decision loss (the loss incurred by the Bayes decision rule $\delta^{*,i}_h(X)$) and the reliability gap (averaged over the possible threshold decision rules).

$$\text{Decision Loss} = \mathbb{E}_X \mathbb{E}_{Y \sim h^*[X]}[\ell^i(X, Y, \delta^{*,i}_h(X))] \quad \text{Reliability Gap} = \frac{1}{|\mathcal{C}|} \sum_{\alpha \in \mathcal{C}} |\gamma(\delta^i_{h,\alpha}, \ell^i)|.$$

Aggregate statistics can be obtained by averaging over all M decision makers, all decision makers who share the same threshold y_0, or all decision makers who share the same cost ratio c.

5.4 Results

Using the MIMIC-III and UCI Protein datasets, we study the effect of recalibration on the reliability gap and the decision loss achieved by decision makers with different decision thresholds and cost ratios. Furthermore, we examine the effect of recalibration on forecasters that output CDFs from simple (Gaussian) and complex (Gaussian-Laplace mixture) model families.

Threshold Calibration Minimizes Reliability Gap Across Decision Thresholds. We evaluate the effect of recalibrating the Gaussian forecaster on decision makers with different decision thresholds. On both datasets, threshold calibration yields the largest decrease in the reliability gap (left plots, Figure 2). Distribution calibration also decreases the reliability gap across decision thresholds, relative to the uncalibrated Gaussian forecaster. Average calibration does not consistently reduce the reliability gap; on the UCI Protein dataset, the reliability gap of the average-calibrated forecaster enjoys a slight decrease at some decision thresholds but is increased at others, relative to the uncalibrated Gaussian forecaster (middle left, Figure 2). Lastly, these calibration methods achieve similar decision loss to the uncalibrated forecaster (right plots, Figure 2). These trends are consistent with the results obtained on the other datasets under the Gaussian forecaster. Threshold calibration outperforms baselines across different decision thresholds under the Gaussian-Laplace forecaster, as well (Appendix A).

Threshold Calibration Minimizes Reliability Gap Across Decision Cost Ratios. Across decision makers with different cost ratios, distribution and threshold calibration reduce the reliability gap relative to the uncalibrated forecaster, with threshold calibration yielding the largest decreases in the reliability gap (left plots, Figure 3). Meanwhile, average calibration does not consistently reduce the reliability gap; on the UCI Protein dataset, it achieves similar reliability gap to the uncalibrated forecaster (middle left, Figure 3). As before, these calibration methods achieve similar decision loss to the uncalibrated forecaster (right plots, Figure 3). These trends are consistent with results obtained on the other datasets under the Gaussian forecaster. Threshold calibration outperforms baselines across different decision cost ratios under the Gaussian-Laplace forecaster, as well (Figure 4).

Distribution Calibration Degrades Performance under Complex Model Families. Forecasters that can output CDFs from more flexible model families (e.g. Gaussian-Laplace mixture distributions)

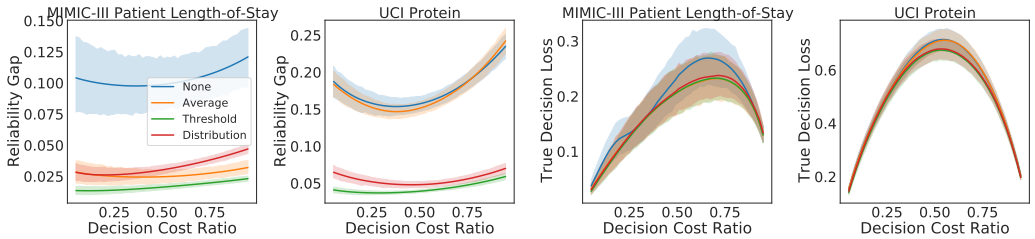

Figure 3: Under the Gaussian forecaster and across different decision cost ratios, threshold calibration reduces the reliability gap on both datasets while average calibration does not reduce the reliability gap on the Protein dataset (**Left, Middle Left**), and all calibration methods yield improved or comparable decision loss compared to the uncalibrated forecaster (**Middle Right, Right**). Error bars represent 95% confidence intervals and are generated over 6 random trials.

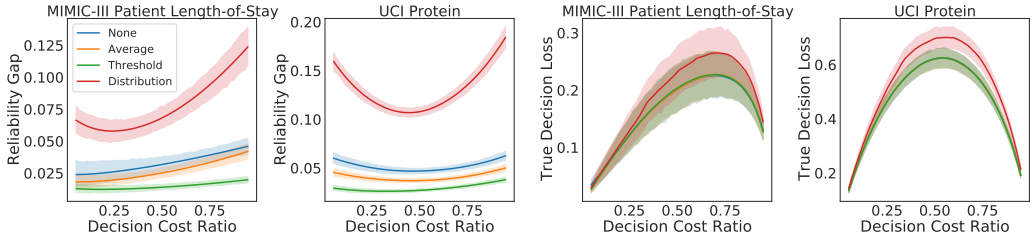

Figure 4: We consider the effect of recalibrating the Gaussian-Laplace forecaster under a range of decision cost ratios. Threshold calibration reduces the reliability gap while distribution calibration can enlarge the reliability gap (**Left, Middle Left**). Average and threshold calibration achieve comparable or lower decision loss as the baseline forecaster, while distribution calibration increases the decision loss. Error bars represent 95% confidence intervals and are generated over 6 random trials.

may be able to better capture the true conditional distribution of Y given x compared to Gaussian forecasters. As a result, we examine the effect of the recalibration procedures when the uncalibrated forecasts follow a more flexible distribution. The uncalibrated Gaussian-Laplace forecaster (Figure 4) yields a smaller reliability gap and smaller decision loss compared to the uncalibrated Gaussian forecaster (Figure 3). Applying threshold calibration to the Gaussian-Laplace forecaster further reduces the reliability gap. However, under the Gaussian-Laplace forecaster, distribution calibration enlarges the size of the reliability gap and increases the decision loss. Although distribution calibration is theoretically guaranteed to minimize the reliability gap, it is challenging to achieve in finite samples without compromising the *sharpness* of the forecasts (in our case, decision loss). So, we find that decision loss and reliability gap increase. We hypothesize that the recalibration dataset may not contain many instances that yield similar distribution parameters, so the recalibration transform does not generalize well to unseen data. We also observe these trends the UCI Crime, UCI Energy, and DHS datasets.

6 Related Work

Forecasting and Decision Making. The connection between forecasts and decision making was first studied in economics [1, 29]. The development of Bayesian decision analysis connected topics of forecasts and decision-based loss functions [10, 4]. Decision-making under uncertainty with probabilistic forecasts was then studied in econometrics [7]. [19] also considers learning regression functions that minimize a decision loss. While [19] focuses on transforming the predicted CDF to a point prediction, our method focuses on transforming the predicted CDF into a new CDF. [19] also requires knowing the loss function to learn the transformation, while our method assumes that the loss function belongs to a commonly used function family (threshold loss functions).

Calibration. Calibration definitions have been studied in the statistics literature [5, 6, 26]. For the regression setting, methods for ensuring that machine learning models satisfy average calibration

have been studied in [23, 8]. In addition, methods for achieving stronger calibration notions have also been introduced such as distribution calibration [31] and individual calibration [34]. Calibration and trustworthy predictions in the medical domain are also studied in [16]. [16] introduces the notion of D-calibration, which is related to our average calibration baseline, but is tailored to the survival analysis task. A perfectly average calibrated prediction function is also D-calibrated, and vice versa.

Multicalibration. Our definition of threshold calibration is most related to the line of work on multicalibration [18, 21]. Given a large collection $\mathcal{G}$ of potentially intersecting groups of the data, a predictor is multicalibrated on $\mathcal{G}$ if it is simultaneously calibrated on every sufficiently large group in $\mathcal{G}$ [18]. Previous works give methods for achieving mean and moment multicalibration for predictor functions. Our iterative procedure for achieving threshold calibration is inspired by methods for achieving multicalibration.

7 Limitations and Societal Impact

Our work demonstrates that certain types of calibration enable decision makers to estimate decision loss before deployment, which should not be confused with enabling decision makers to make optimal decisions. For example, a forecaster that always outputs the marginal distribution of Y is threshold-calibrated but likely incurs high decision loss. Furthermore, posthoc recalibration is limited by the quality of the baseline model. If the baseline model outputs the marginal distribution of Y, then it is already threshold-calibrated but likely is not useful for decision making. Applying our threshold calibration method will not offer any benefit in this case.

Also, our work assumes that predictions of Y do not affect the true label Y. However, when predictions are used to make decisions, they can often influence the outcome they aim to predict [27]. Our work does not account for these performative effects, so the decision loss may not be accurately estimated in these settings. Future work could focus on developing calibration procedures that enable forecasters to be robust to such distribution shifts. In addition, we specifically focus on binary-action threshold decisions. Future work may generalize our results to the setting where decision makers have loss functions involving multiple thresholds and multiple actions.

There is a potential for negative societal impact if threshold calibration is incompatible with fairness criteria. Nevertheless, we note that the perfect predictor (that predicts the true conditional probability) satisfies our calibration definition. Consequently, if the perfect predictor satisfies some fairness notion (such as group calibration), then our calibration definition is also compatible with that fairness notion. Note that the perfect predictor does not satisfy a fairness notion called demographic parity, hence our calibration definition is not compatible with demographic parity either.

8 Conclusion

We show that a threshold-calibrated forecaster theoretically guarantees accurate decision loss estimation under threshold decision losses and threshold decision rules. We provide an iterative procedure for achieving threshold calibration and show that in practice it minimizes the reliability gap relative to baselines without compromising the forecaster's decision loss. Such estimates permit decision makers to reason about the consequences of their decisions prior to deployment.

Acknowledgements

RS is supported in part by a NSF GRFP under grant number DGE-1656518. SZ is supported in part by a JP Morgan fellowship and a Qualcomm innovation fellowship. SE is supported in part by NSF(#1651565, #1522054, #1733686), ONR (N000141912145), AFOSR (FA95501910024), ARO (W911NF-21-1-0125) and Sloan Fellowship. We are grateful for Rishi Bommasani, Kristy Choi, Matthew Jörke, Judy Shen, Rui Shu, Fan-Yun Sun, Rohan Taori, Ke Alexander Wang, Rose Wang, and Henry Zhu for insightful discussions.

References

[1] H. theil. economic forecasts and policy. assisted by j.s. cramer, h. moerman, a. russchen. contributions to economic analysis, nr xv. amsterdam, north-holland publishing company, 1958,

xxxi p. 562 p., fl. 50—. *Bulletin de l'Institut de recherches économiques et sociales*, 25(2): 169–169, 1959. doi: 10.1017/S1373971900078951.

[2] Alexander Amini, Wilko Schwarting, Ava Soleimany, and Daniela Rus. Deep evidential regression. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 14927–14937. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/aab085461de182608ee9f607f3f7d18f-Paper.pdf.

[3] S. Barnes, Eric Hamrock, Matthew F. Toerper, S. Siddiqui, and S. Levin. Real-time prediction of inpatient length of stay for discharge prioritization. *Journal of the American Medical Informatics Association : JAMIA*, 23 e1:e2–e10, 2016.

[4] James O. Berger and James O. Berger. *Statistical decision theory and Bayesian analysis*. Springer-Verlag, New York, 1985. ISBN 0387960988 9780387960982 3540960988 9783540960980. URL http://www.amazon.com/Statistical-Decision-Bayesian-Analysis-Statistics/dp/0387960988/ref=sr_1_11?ie=UTF8&qid=1403880466&sr=8-11&keywords=Bayesian+statistics.

[5] GLENN W. BRIER. Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, 78(1):1 – 3, 1950. doi: 10.1175/1520-0493(1950)078<0001:VOFEIT>2.0.CO;2. URL https://journals.ametsoc.org/view/journals/mwre/78/1/1520-0493_1950_078_0001_vofeit_2_0_co_2.xml.

[6] Nicolo Cesa-Bianchi and Gabor Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, USA, 2006. ISBN 0521841089.

[7] Gary Chamberlain. Econometrics and decision theory. *Journal of Econometrics*, 95:255–283, 2000.

[8] Peng Cui, Wenbo Hu, and Jun Zhu. Calibrated reliable regression using maximum mean discrepancy. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 17164–17175. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/c74c4bf0dad9cbae3d80faa054b7d8ca-Paper.pdf.

[9] Murray Dale, Jon Wicks, Ken Mylne, Florian Pappenberger, Stefan Laeger, and Steve Taylor. Probabilistic flood forecasting and decision-making: An innovative risk-based approach. *Natural Hazards*, 70, 11 2014. doi: 10.1007/s11069-012-0483-z.

[10] Morris H. DeGroot. *Optimal statistical decisions*. McGraw-Hill, New York, NY [u.a], 1970. ISBN 0070162425. URL http://gso.gbv.de/DB=2.1/CMD?ACT=SRCHA&SRT=YOP&IKT=1016&TRM=ppn+021834997&sourceid=fbw_bibsonomy.

[11] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

[12] Tilmann Gneiting, Fadoua Balabdaoui, and Adrian E. Raftery. Probabilistic forecasts, calibration and sharpness. *Journal of the Royal Statistical Society Series B*, 69(2):243–268, 2007. URL https://EconPapers.repec.org/RePEc:bla:jorssb:v:69:y:2007:i:2:p:243-268.

[13] Ary L Goldberger, Luis AN Amaral, Leon Glass, Jeffrey M Hausdorff, Plamen Ch Ivanov, Roger G Mark, Joseph E Mietus, George B Moody, Chung-Kang Peng, and H Eugene Stanley. Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals. *Circulation*, 101(23):e215–e220, 2000.

[14] Clive W.J. Granger and Mark J. Machina. Forecasting and decision theory. In G. Elliott, C. Granger, and A. Timmermann, editors, *Handbook of Economic Forecasting*, volume 1 of *Handbook of Economic Forecasting*, chapter 2, pages 81–98. Elsevier, 2006. URL https://ideas.repec.org/h/eee/ecofch/1-02.html.

[15] Margaret Grosh, Carlo del Ninno, Emil Tesliuc, and Azedine Ouerghi. *For Protection and Promotion: The Design and Implementation of Effective Safety Nets*. The World Bank, 2008. URL https://EconPapers.repec.org/RePEc:wbk:wbpubs:6582.

[16] Humza Haider, Bret Hoehn, Sarah Davis, and Russell Greiner. Effective ways to build and evaluate individual survival distributions. *Journal of Machine Learning Research*, 21(85):1–63, 2020. URL http://jmlr.org/papers/v21/18-772.html.

[17] H. Harutyunyan, Hrant Khachatrian, David C. Kale, and A. Galstyan. Multitask learning and benchmarking with clinical time series data. *Scientific Data*, 6, 2019.

[18] Ursula Hebert-Johnson, Michael Kim, Omer Reingold, and Guy Rothblum. Multicalibration: Calibration for the (Computationally-identifiable) masses. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1939–1948. PMLR, 10–15 Jul 2018. URL http://proceedings.mlr.press/v80/hebert-johnson18a.html.

[19] José Hernandez-Orallo. Probabilistic reframing for cost-sensitive regression. *ACM Trans. Knowl. Discov. Data*, 8(4), August 2014. ISSN 1556-4681. doi: 10.1145/2641758. URL https://doi.org/10.1145/2641758.

[20] Alistair EW Johnson, Tom J Pollard, Lu Shen, Li-wei H Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3:160035, 2016.

[21] Christopher Jung, Changhwa Lee, Mallesh M. Pai, Aaron Roth, and Rakesh Vohra. Moment multicalibration for uncertainty estimation, 2020.

[22] Ranganath Krishnan and Omesh Tickoo. Improving model calibration with accuracy versus uncertainty optimization. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 18237–18248. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/d3d9446802a44259755d38e6d163e820-Paper.pdf.

[23] Volodymyr Kuleshov, Nathan Fenner, and Stefano Ermon. Accurate uncertainties for deep learning using calibrated regression. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 2801–2809. PMLR, 2018. URL http://proceedings.mlr.press/v80/kuleshov18a.html.

[24] Meelis Kull, Telmo de Menezes e Silva Filho, and Peter A. Flach. Beta calibration: a well-founded and easily implemented improvement on logistic calibration for binary classifiers. In Aarti Singh and Xiaojin (Jerry) Zhu, editors, *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 April 2017, Fort Lauderdale, FL, USA*, volume 54 of *Proceedings of Machine Learning Research*, pages 623–631. PMLR, 2017. URL http://proceedings.mlr.press/v54/kull17a.html.

[25] Ali Malik, Volodymyr Kuleshov, Jiaming Song, Danny Nemer, Harlan Seymour, and Stefano Ermon. Calibrated model-based deep reinforcement learning. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 4314–4323. PMLR, 2019. URL http://proceedings.mlr.press/v97/malik19a.html.

[26] Allan H. Murphy. A new vector partition of the probability score. *Journal of Applied Meteorology and Climatology*, 12(4):595 – 600, 1973. doi: 10.1175/1520-0450(1973)012<0595:ANVPOT>2.0.CO;2. URL https://journals.ametsoc.org/view/journals/apme/12/4/1520-0450_1973_012_0595_anvpot_2_0_co_2.xml.

[27] Juan Perdomo, Tijana Zrnic, Celestine Mendler-Dünner, and Moritz Hardt. Performative prediction. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 7599–7609. PMLR, 13–18 Jul 2020. URL http://proceedings.mlr.press/v119/perdomo20a.html.

[28] Geoff Pleiss, Manish Raghavan, Felix Wu, Jon Kleinberg, and Kilian Q Weinberger. On fairness and calibration. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/b8b9c74ac526fffbeb2d39ab038d1cd7-Paper.pdf.

[29] Robert W. Rudd. Theil, henri, applied economic forecasting, chicago, rand mcnally ... company, 1966, xxv + 474 pp. ($14.00). *American Journal of Agricultural Economics*, 49(1_Part_I):241–243, 1967. doi: https://doi.org/10.2307/1237096. URL https://onlinelibrary.wiley.com/doi/abs/10.2307/1237096.

[30] Evan Sheehan, Chenlin Meng, Matthew Tan, Burak Uzkent, Neal Jean, Marshall Burke, David Lobell, and Stefano Ermon. Predicting economic development using geolocated wikipedia articles. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, page 2698–2706, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450362016. doi: 10.1145/3292500.3330784. URL https://doi.org/10.1145/3292500.3330784.

[31] Hao Song, Tom Diethe, Meelis Kull, and Peter A. Flach. Distribution calibration for regression. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 5897–5906. PMLR, 2019. URL http://proceedings.mlr.press/v97/song19a.html.

[32] Christopher Yeh, Anthony Perez, Anne Driscoll, George Azzari, Zhongyi Tang, David Lobell, Stefano Ermon, and Marshall Burke. Using publicly available satellite imagery and deep learning to understand economic well-being in africa. *Nature Communications*, 11(1), 5 2020. ISSN 2041-1723. doi: 10.1038/s41467-020-16185-w. URL https://www.nature.com/articles/s41467-020-16185-w.

[33] Weiran Yuchi, Jiayun Yao, Kathleen E. McLean, Roland Stull, Radenko Pavlovic, Didier Davignon, Michael D. Moran, and Sarah B. Henderson. Blending forest fire smoke forecasts with observed data can improve their utility for public health applications. *Atmospheric Environment*, 145:308–317, 2016. ISSN 1352-2310. doi: https://doi.org/10.1016/j.atmosenv.2016.09.049. URL https://www.sciencedirect.com/science/article/pii/S1352231016307592.

[34] Shengjia Zhao, Tengyu Ma, and Stefano Ermon. Individual calibration with randomized forecasting. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 11387–11397. PMLR, 13–18 Jul 2020. URL http://proceedings.mlr.press/v119/zhao20e.html.

Checklist

The checklist follows the references. Please read the checklist guidelines carefully for information on how to answer these questions. For each question, change the default **[TODO]** to [Yes] , [No] , or [N/A] . You are strongly encouraged to include a **justification to your answer**, either by referencing the appropriate section of your paper or providing a brief inline description. For example:

- Did you include the license to the code and datasets? [Yes] See Appendix A.
- Did you include the license to the code and datasets? [No] The code and the data are proprietary.
- Did you include the license to the code and datasets? [N/A]

Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all authors...

(a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]

(b) Did you describe the limitations of your work? [Yes] See Section 7.

(c) Did you discuss any potential negative societal impacts of your work? [Yes] See Section 7.

(d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]

2. If you are including theoretical results...

(a) Did you state the full set of assumptions of all theoretical results? [Yes]

(b) Did you include complete proofs of all theoretical results? [Yes] See Appendix B.

3. If you ran experiments...

(a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]

(b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Appendix A

(c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See Section 5 and Appendix A

(d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Appendix A.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

(a) If your work uses existing assets, did you cite the creators? [Yes] See Appendix A.

(b) Did you mention the license of the assets? [Yes] See Appendix A.

(c) Did you include any new assets either in the supplemental material or as a URL? [Yes] See Appendix A.

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] We did not directly obtain data from anyone but we used publicly available datasets.

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] The creators of the MIMIC-III dataset, which we use, deanonymized the data to remove any personally identifiable information. To the best of our knowledge, there is no other potential source of personally identifiable or offensive content in the data we use.

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A] We did not do human subject research.

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A] We did not do human subject research.

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A] We did not do human subject research.

A Experimental Setup and Additional Results

A.1 Reproducibility

We provide a link to our code below. The code includes scripts for downloading the UCI regression datasets. Accessing the MIMIC-III dataset requires an ethics training course and permissions [20], so we do not provide the dataset or download information in the code.

- https://drive.google.com/file/d/12Qh1AWsJcx6UzrRAVYPAYPNemRBj7500/view?usp=sharing.

A.2 Baseline Forecasters

In our experiments, a forecaster is trained on data $\{(x_i, y_i)\}_{i=1}^n$. We assume the labels y_i are drawn i.i.d. from a distribution with a parameters θ_i. We consider two types of predictive distributions, a unimodal Gaussian distribution and a mixture of a Gaussian and a Laplace distribution. For a Gaussian distribution, the distribution parameters θ_i consists of a mean μ_i and standard deviation σ_i. For a mixture of a Laplace and Gaussian distribution, the distribution parameters θ_i consist of the weight assigned to the Gaussian component w_i, the Gaussian mean μ_i and standard deviation σ_i, and the Laplace location m_i and scale b_i.

The forecaster is a neural network h_w where w denotes the parameters of the network. The network takes x as input. For the Gaussian forecaster, the network outputs the parameters of a Gaussian distribution (2 parameters). For the Gaussian-Laplace forecaster, the network takes x as input and outputs the parameters of a Gaussian-Laplace distribution (5 parameters). The network can be trained by using negative log likelihood of the predictive distribution as the loss function.

A.3 Recalibration Procedure

The posthoc recalibration transforms are fit using a recalibration dataset. We detail the recalibration procedure for each type of calibration.

A.3.1 Average.

To enforce average calibration, we use the method defined in [23], using the recalibration dataset to fit a single isotonic regression with linear interpolation.

A.3.2 Threshold

We use a finite sample version of the method described in Section 4 to enforce threshold calibration. We run the algorithm for $T = 40$ iterations for all datasets. We give a detailed outline of the algorithm we use.

Algorithm 2: Threshold Recalibration

Input: Uncalibrated forecaster $h : \mathcal{X} \to \mathcal{F}(\mathcal{Y})$, recalibration dataset $\mathcal{D} = \{x_i, y_i\}_{i=1}^n$, discretization parameter $K \in \mathbb{Z}_+$, number of iterations T
Output: A threshold-calibrated model $h^T : \mathcal{X} \to \mathcal{F}(\mathcal{Y})$.
$m, M \leftarrow \inf_{i\in[n]} y_i, \sup_{i\in[n]} y_i$
$\mathcal{Y} \leftarrow \{m + \frac{(M-m)j}{K} \mid j = 1, 2, \dots K\}$
$\mathcal{Q} \leftarrow \{\frac{j}{K} \mid j = 1, 2, \dots K\}$
$h^0 \leftarrow h$
for $t = 1, 2, 3, \dots T$ **do**
    Select the $(y_0^t, \alpha^t,)$ that yields the highest TCE.
    $(y_0^t, \alpha^t) \leftarrow \arg\sup_{(y_0,\alpha)\in\mathcal{Y}\times\mathcal{Q}} \widehat{\text{TCE}}(h^{t-1}, y_0, \alpha)$
    Compute h^t applying Algorithm 3 with the arguments $h^{t-1}, y_0^t, \alpha^t, \mathcal{D}$
end
return h^T

Algorithm 3: Recalibration at Single Threshold-Quantile Pair

Input: Uncalibrated forecaster $h : \mathcal{X} \to \mathcal{F}(\mathcal{Y})$, recalibration dataset $\mathcal{D} = \{x_i, y_i\}_{i=1}^n$, threshold $y_0 \in \mathbb{R}$, quantile $\alpha \in [0, 1]$, discretization parameter $K \in \mathbb{Z}_+$.
Output: A model $h : \mathcal{X} \to \mathcal{F}(\mathcal{Y})$ that is threshold-calibrated at a threshold y_0 and quantile α.
Partition the data based on whether $h[x_i](y_0) \leq \alpha$:
$\mathcal{I}_1 \leftarrow \{i \in [n] \mid h[x_i](y_0) \leq \alpha\}$.
$\mathcal{I}_2 \leftarrow [n] \setminus \mathcal{I}_1$.
Learn recalibration functions $\mathcal{R}_k$ for each $\mathcal{I}_k$:
for $k = 1, 2$ **do**
    Create recalibration dataset $\mathcal{D}_k \leftarrow \{h[x_i](y_i), \hat{P}_k(h[x_i](y_i))\}_{i \in \mathcal{I}_k}$, where $\hat{P}_k(p) \leftarrow |\{i \in \mathcal{I}_k \mid h[x_i](y_i) \leq p\}|/|\mathcal{I}_k|$.
    Train a model R_k on $\mathcal{D}_k$ (e.g. isotonic regression).
$h[x] \leftarrow \begin{cases} R_1(h[x]) & \text{if } h[x](y_0) \leq \alpha \\ R_2(h[x]) & \text{otherwise} \end{cases}$
return h

Recalibration at a Single Threshold-Quantile Pair (Algorithm 3). Given an uncalibrated model h, a recalibration dataset $\mathcal{D}$, and a discretization parameter K, we propose a simple recalibration procedure for enforcing calibration for a single threshold-quantile pair y_0, α (Algorithm 3). We give an overview of the algorithm. First, we partition the recalibration samples into two bins, $\mathcal{I}_1$ and $\mathcal{I}_2$, based on whether the predicted CDF value $h[x](y_0)$ is greater than α. Next, we learn a recalibration transform R_k for $\mathcal{I}_k$ using a method similar to [23]. To ensure that the recalibrated forecaster outputs valid CDFs, we require that each $R_k : [0, 1] \to [0, 1]$ and is monotonically increasing. For a particular sample (x, y), the appropriate recalibration transform R_k to apply depends on whether $h[x](y_0)$ is greater than α.

A.3.3 Distribution.

To enforce distribution calibration, we construct p-dimensional grid where p is the number of parameters in the forecaster's model family. For Gaussian distributions, we have $p = 2$. For Gaussian-Laplace mixture distributions, we have $p = 5$. We set the grid boundaries by computing the range of each distribution parameter on the validation set. We uniformly partition each axis of the grid into K bins. Each validation sample is sorted into a single grid cell based on the predicted distribution parameters. We fit an isotonic regression model (with linear interpolation) as in [23] using the validation samples that fall into a particular grid cell. For evaluation, we sort the test examples into the appropriate grid cell and apply the corresponding recalibration model. We set the number of bins for each parameter to $K = 20$.

A.4 Calibration Metrics

The expected calibration error (ECE) is used to measure deviations from average calibration [23]. It is defined as follows

$$\text{ECE}(h) = \int_{c \in [0,1]} |\Pr[h[X](Y) \leq c] - c| \quad dc.$$

We contrast this definition with TCE, threshold calibration error, which measures deviations from threshold calibration. Smaller ECE implies better average calibration. Smaller TCE implies better threshold calibration.

A.5 Hospital Scheduling Decisions on MIMIC-III

A.5.1 Dataset

Medical Information Mart for Intensive Care III (MIMIC-III) is a freely accessible medical database of critically ill patients admitted to the intensive care unit (ICU) at Beth Israel Deaconess Medical Center (BIDMC) from 2001 to 2012 [20, 13]. During that time, BIDMC switched clinical information

systems from Carevue (2001-2008) to Metavision (2008-2012). To ensure data consistency, only data archived via the Metavision system was used in the dataset.

Feature Selection. We select the same patient features and imputed values as in [17]. A total of 17 variables were extracted from the chartevents table to include in the dataset - capillary refill, blood pressure (systolic, diastolic, and mean), fraction of inspired oxygen, Glasgow Coma Score (eye opening response, motor response, verbal response, and total score), serum glucose, heart rate, respiratory rate, oxygen saturation, respiratory rate, temperature, weight, and arterial pH. For each unique ICU stay, values were extracted for the first 24 hours upon admission to the ICU and averaged. Normal values were imputed for missing variables as shown in Table 1. There are 26089 unique ICU stays in the dataset. The final dataset consisted of the total length of ICU stay and the mean value for each of the 17 variables across the first 24 hours.

Variable	MIMIC-III item ids from chartevents table	Imputed value
Capillary refll rate	(223951, 224308)	0
Diastolic blood pressure	(220051, 227242, 224643, 220180, 225310)	59.0
Systolic blood pressure	(220050, 224167, 227243, 220179, 225309)	118.0
Mean blood pressure	(220052, 220181, 225312)	77.0
Fraction inspired oxygen	(223835)	0.21
GCS eye opening	(220739)	4
GCS motor response	(223901)	6
GCS verbal response	(223900)	5
GCS total	(220739 + 223901 + 223900)	15
Glucose	(228388, 225664, 220621, 226537)	128.0
Heart Rate	(220045)	86
Height	(226707, 226730)	170.0
Oxygen saturation	(220227, 220277, 228232)	98.0
Respiratory rate	(220210, 224688, 224689, 224690)	19
Temperature	(223761, 223762)	97.88
Weight	(224639, 226512, 226531)	178.6
pH	(223830)	7.4

Table 1: Variables included in dataset

Dataset Splits. For each of 6 random seeds $[0, 1, 2, 3, 4, 5]$, we generate different dataset splits. Given a random seed, we randomly split off 30% of the original dataset to use as the test set. The remaining dataset are further partitioned into a validation set and training set. The validation set consists of 10% of the remaining data.

A.5.2 Toy Example.

Setup. The decision task of interest in Figure 1 is identifying patients with LOS longer than 3.5 days. The optimal decision rule is $\phi(y) = \mathbb{I}(y \geq y_0)$. In the absence of true LOS values, the forecast-based decision rule $\delta(X) = \mathbb{I}(h[x](y_0) \leq \alpha)$ where $\alpha \in [0, 1]$. To evaluate decision rules, we consider a loss function of the form $\ell(x, y, a) = 5\mathbb{I}(a = 1, y < y_0) + 5\mathbb{I}(a = 0, y \geq y_0)$. So, false positives and false negatives incur equal cost and right decisions incur no cost.

Forecaster Training Procedure. For each dataset split, we train an average-calibrated forecaster and a threshold-calibrated forecaster.

To obtain a forecaster that obtains perfect average calibration, we train a neural network with 3 hidden layers of 100 units and ReLU activation on training split of the dataset with ECE as the loss function. We check the validation ECE at each epoch and save the model that obtains lowest validation ECE.

To obtain a threshold-calibrated forecaster, we train a neural network with 3 hidden layers of 100 units and ReLU activation on the training split of the dataset with TCE as the loss function. We use a discretization parameter of $K = 100$ for training the forecaster. We check the validation TCE at each epoch and save the model that obtains lowest validation TCE.

We train both forecasters for 300 iterations.

Results. Since the forecasters are only trained with a calibration objective, we do not expect them to provide accurate decisions. Nevertheless, we assess the *reliability* of the forecasters by evaluating how

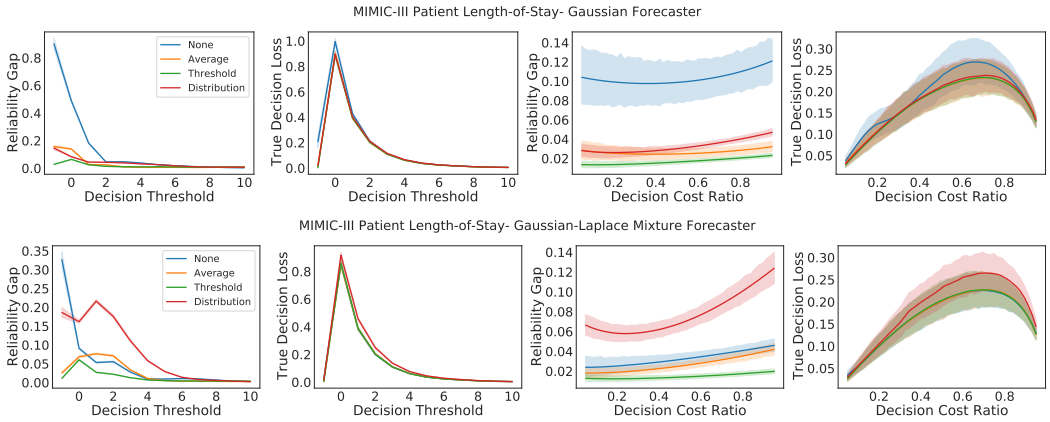

Figure 5: Hospital Scheduling Decisions with Patient Length-of-Stay Forecaster. We plot the decision loss and the reliability gap achieved across decision costs and decision thresholds with various recalibration methods. Threshold calibration achieves the smallest reliability gap among baselines across different decision thresholds and decision cost ratios without compromising the decision loss. Error bars denote 95 % confidence intervals computed over 6 random trials.

well they can predict the true decision loss. The average-calibrated forecaster appears to accurately predict the decision loss at $\alpha = 0.5$, but underestimates the loss at $\alpha > 0.5$ and overestimates the loss at $\alpha < 0.5$.

Calibration Method	**Reliability Gap**	**TCE**	**ECE**
Average-Calibrated Forecaster	0.700 ± 0.342	0.176 ± 0.013	0.006 ± 0.002
Threshold-Calibrated Forecaster	0.027 ± 0.020	0.038 ± 0.008	0.001 ± 0.002

Table 2: Patient Length-of-Stay Forecasters. The average-calibrated forecaster has a large reliability gap despite achieving near perfect ECE.

A.5.3 Recalibration Experiment Details.

Baseline Models. We train a neural network with 3 hidden layers of 100 units with ReLU activation. The number of inputs to the network is the dimension of the features and the number of outputs of network is the number of parameters of the outputted distribution. This is 2 parameters in the case of outputting Gaussian distributions and 5 parameters in the case of outputting a mixture of a Gaussian and Laplace distribution.

Baseline Training Procedure. The baseline models are trained for a maximum of 100 epochs with batch size equal to 128 and we use the Adam optimizer with learning rate 1e-3. Each epoch we check the loss obtained on the validation set and select the model that minimizes the loss on the validation set.

Recalibration Procedure. We use the validation set to learn the recalibration transform.

A.5.4 Recalibration Results.

In Table 3, we show the mean reliability gap and mean decision loss obtained over all M decision makers and TCE and ECE of the forecaster on the MIMIC-III dataset. The standard deviation is computed over 6 randomized trials. The findings are consistent with the findings reported in Section 5; we observe that threshold calibration minimizes the reliability gap and obtains the lowest TCE among the baselines without compromising the decision loss. Although average calibration achieves the lowest ECE across baselines, it does not consistently improve the reliability gap across different decision thresholds. In addition, distribution calibration can increase the reliability gap and decision loss when the model family of the forecaster is more complex (more flexible).

Forecaster	Method	**Reliability Gap**	**Decision Loss**	**TCE**	**ECE**
Gaussian	None	0.103 ± 0.019	0.189 ± 0.008	0.227 ± 0.017	0.115 ± 0.01
	Average	0.027 ± 0.005	0.17 ± 0.005	0.093 ± 0.019	$\mathbf{0.009 \pm 0.004}$
	Threshold	$\mathbf{0.017 \pm 0.006}$	$\mathbf{0.17 \pm 0.005}$	$\mathbf{0.055 \pm 0.013}$	0.011 ± 0.003
	Distribution	0.033 ± 0.006	0.173 ± 0.006	0.061 ± 0.008	0.011 ± 0.004
Gaussian-Laplace	None	0.033 ± 0.002	$\mathbf{0.165 \pm 0.003}$	0.082 ± 0.01	0.034 ± 0.003
	Average	0.027 ± 0.005	$\mathbf{0.165 \pm 0.003}$	0.052 ± 0.007	$\mathbf{0.009 \pm 0.003}$
	Threshold	$\mathbf{0.015 \pm 0.005}$	0.166 ± 0.003	$\mathbf{0.048 \pm 0.014}$	0.011 ± 0.002
	Distribution	0.077 ± 0.013	0.188 ± 0.006	0.115 ± 0.02	0.032 ± 0.009

Table 3: Recalibration results for Patient Length-of-Stay Forecasting on MIMIC-III dataset. We observe that threshold calibration procedure decreases the reliability gap.

A.6 Resource Allocation Decisions on Demographic and Health Survey

A.6.1 Dataset

We use the satellite images and asset wealth data for African countries of Tanzania, Malawi, Mozambique, Uganda, Rwanda, Zimbabwe from the Demographic and Health Surveys (DHS) from 2009-2011 [30]. We use the nightlight bands of the satellite images. The dataset contains 4191 samples.

Dataset Splits. For each of 6 random seeds $[0, 1, 2, 3, 4, 5]$, we generate different dataset splits. Given a random seed, we randomly split off 30% of the original dataset to use as the test set. The remaining dataset are further partitioned into a validation set and training set. The validation set consists of 10% of the remaining data.

A.6.2 Recalibration Experiment Details

Baseline Models. The neural network model that we use is a pretrained Resnet18 architecture from the Pytorch model zoo, which is adjusted to have grayscale inputs. The input shape $255 \times 255 \times 1$ and the number of outputs of network is the number of parameters of the predicted distribution.

Baseline Training Procedure. The baseline models are trained for a maximum of 100 epochs with batch size equal to 32 and we use the Adam optimizer with learning rate 1e-3. Each epoch we check the loss obtained on the validation set and select the model that minimizes the loss on the validation set.

Recalibration Procedure. Due to the small size of the dataset, the training set and validation set are used to train the recalibration transform.

A.6.3 Recalibration Results.

In Table 4, we show the mean reliability gap and mean decision loss obtained over all M decision makers and TCE and ECE of the forecaster on the DHS Asset Wealth dataset. The standard deviation is computed over 6 randomized trials. The findings are consistent with the findings reported in Section 5; we observe that threshold calibration minimizes the reliability gap and obtains the lowest TCE among the baselines without compromising the decision loss.

A.7 UCI Regression Datasets

A.7.1 Datasets

We use 4 UCI regression datasets. Three of the datasets, Protein, Energy, and Naval, are large and contain 45730, 19735, and 11934 samples respectively. The smaller dataset, Crime, contains 1994 samples.

Dataset Splits. For each of 6 random seeds $[0, 1, 2, 3, 4, 5]$, we generate different dataset splits. Given a random seed, we randomly split off 30% of the original dataset to use as the test set. The remaining dataset are further partitioned into a validation set and training set. The validation set consists of 10% of the remaining data.

Forecaster	Method	**Reliability Gap**	**Decision Loss**	**TCE**	**ECE**
Gaussian	None	0.086 ± 0.017	0.294 ± 0.009	0.095 ± 0.026	0.047 ± 0.016
	Average	0.057 ± 0.007	0.291 ± 0.011	0.068 ± 0.015	$\mathbf{0.014 \pm 0.005}$
	Threshold	$\mathbf{0.033 \pm 0.007}$	$\mathbf{0.287 \pm 0.007}$	$\mathbf{0.048 \pm 0.013}$	0.014 ± 0.006
	Distribution	0.045 ± 0.019	0.295 ± 0.009	0.064 ± 0.027	0.015 ± 0.008
Gaussian-Laplace	None	0.053 ± 0.015	0.288 ± 0.003	0.07 ± 0.022	0.03 ± 0.015
	Average	0.037 ± 0.01	0.287 ± 0.004	0.052 ± 0.021	$\mathbf{0.013 \pm 0.005}$
	Threshold	$\mathbf{0.032 \pm 0.005}$	$\mathbf{0.286 \pm 0.004}$	$\mathbf{0.045 \pm 0.01}$	$\mathbf{0.013 \pm 0.005}$
	Distribution	0.062 ± 0.016	0.305 ± 0.004	0.084 ± 0.028	0.021 ± 0.014

Table 4: Recalibration results for DHS Survey. We observe that threshold calibration procedure improves the reliability gap.

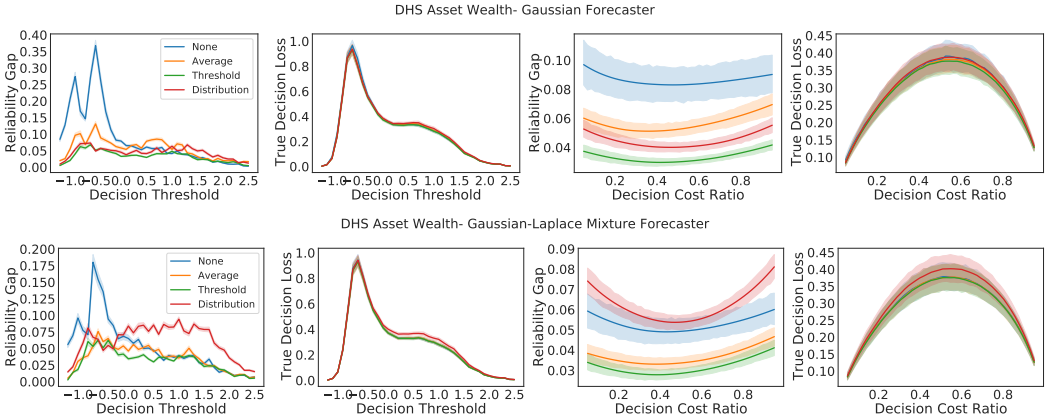

Figure 6: Resource allocation decisions on the DHS Asset Wealth dataset. We plot the decision loss and the reliability gap achieved across decision costs and decision thresholds with various recalibration methods. Threshold calibration achieves the smallest reliability gap among baselines across different decision thresholds and decision cost ratios without compromising the decision loss. Error bars denote 95 % confidence intervals computed over 6 random trials.

A.7.2 Recalibration Experiment Details.

Baseline Models. We train a neural network with 3 hidden layers of 100 units with ReLU activation. The number of inputs to the network is the dimension of the features and the number of outputs of network is the number of parameters of the outputted distribution. This is 2 parameters in the case of outputting Gaussian distributions and 5 parameters in the case of outputting a mixture of a Gaussian and Laplace distribution.

Baseline Training Procedure. The baseline models are trained for a maximum of 100 epochs with batch size equal to 128 and we use the Adam optimizer with learning rate 1e-3. Each epoch we check the loss obtained on the validation set and select the model that minimizes the loss on the validation set.

Recalibration Procedure. For the larger datasets (Protein, Naval, Energy), we use the validation set for recalibration to avoid overfitting. For the small dataset (Crime), we combine the training and validation set for recalibration.

A.7.3 Recalibration Results.

We show two sets of results. The first includes results with the forecaster that outputs Gaussian distributions in Table 5. The second includes results with the forecaster that outputs Gaussian-Laplace mixture distributions in Table 6. As described in Section 5, we observe that using a forecaster with a more flexible model family (Gaussian-Laplace) can decrease the decision loss and reliability gap of the uncalibrated model. Threshold calibration can further decrease the reliability gap of these models,

while distribution calibration is challenging to achieve and may increase the reliability gap and the decision loss.

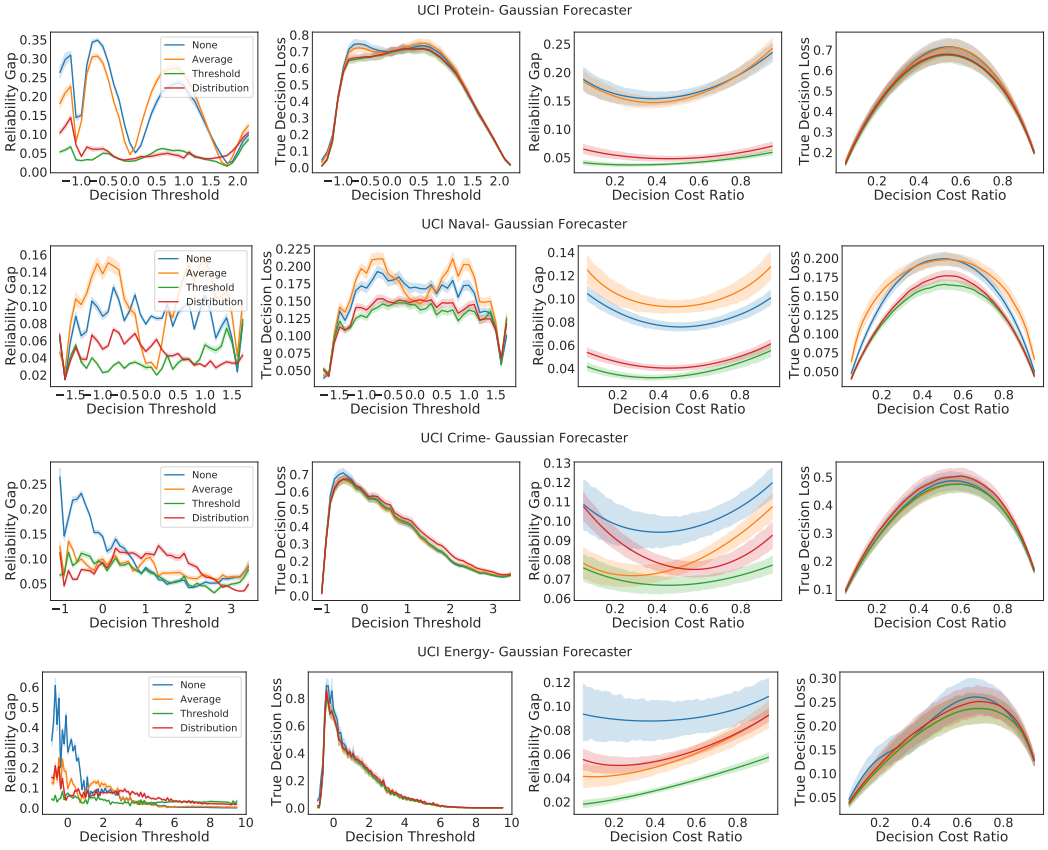

Figure 7: UCI Protein, Naval, Crime, Energy results with Gaussian forecaster. We plot the decision loss and the reliability gap achieved across decision costs and decision thresholds with various recalibration methods. Threshold calibration results in consistent improvements across decision thresholds and decision costs compared to other baselines under the Gaussian forecaster.

B Proofs

B.1 Justification for Threshold Decision Rules

We justify the assumption (from Section 2.3) that a decision maker with a threshold loss function and a decision rule on the forecasted CDFs will restrict the space of decision rules they consider to threshold decision rules on the forecasted CDFs.

Suppose a decision-maker has a threshold loss function and a binary action space $\mathcal{A}$.

$$\ell(x, y, a) = \sum_{i \in \mathcal{A}} \mathbb{I}(y \geq y_0, a = i) c_{1,i} + \sum_{i \in \mathcal{A}} \mathbb{I}(y < y_0, a = i) c_{0,i}.$$

We can compute that the expected loss under the forecasted distribution as follows.

$$\begin{aligned}\mathbb{E}_X \mathbb{E}_{\tilde{Y} \sim h[X]}[\ell(X, \tilde{Y}, \delta(X))] =& \mathbb{E}_X[\mathbb{I}(\delta(X) = 1)(\Pr(\tilde{Y} \geq y_0 \mid X) c_{11} + \Pr(\tilde{Y} < y_0 \mid X) c_{01}) \\ &+ \mathbb{I}(\delta(X) = 0)(\Pr(\tilde{Y} \geq y_0 \mid X) c_{10} + \Pr(\tilde{Y} < y_0 \mid X) c_{00})]\end{aligned}$$

The decision maker minimizes their expected decision loss with respect to the forecasted distribution if $\delta(X) = 1$ when

$$(\Pr(\tilde{Y} \geq y_0 \mid X) c_{11} + \Pr(\tilde{Y} < y_0 \mid X) c_{01} \leq (\Pr(\tilde{Y} \geq y_0 \mid X) c_{10} + \Pr(\tilde{Y} < y_0 \mid X) c_{00}).$$

Dataset	Method	Reliability Gap	Decision Loss	TCE	ECE
Protein (n=45730)	None	0.175 ± 0.012	0.530 ± 0.007	0.123 ± 0.007	0.032 ± 0.005
	Average	0.172 ± 0.005	0.528 ± 0.006	0.104 ± 0.007	$\mathbf{0.005 \pm 0.001}$
	Threshold	$\mathbf{0.043 \pm 0.004}$	$\mathbf{0.506 \pm 0.005}$	$\mathbf{0.047 \pm 0.006}$	0.005 ± 0.001
	Distribution	0.055 ± 0.024	0.511 ± 0.008	0.058 ± 0.010	0.007 ± 0.003
Energy (n=19735)	None	0.092 ± 0.021	0.189 ± 0.021	0.178 ± 0.030	0.081 ± 0.019
	Average	0.055 ± 0.007	0.176 ± 0.017	0.071 ± 0.003	$\mathbf{0.007 \pm 0.002}$
	Threshold	$\mathbf{0.035 \pm 0.003}$	$\mathbf{0.175 \pm 0.017}$	$\mathbf{0.053 \pm 0.007}$	0.010 ± 0.003
	Distribution	0.063 ± 0.013	0.184 ± 0.018	0.121 ± 0.015	0.012 ± 0.004
Naval (n=11934)	None	0.085 ± 0.006	0.152 ± 0.024	0.245 ± 0.031	0.115 ± 0.014
	Average	0.103 ± 0.016	0.162 ± 0.028	0.068 ± 0.011	0.020 ± 0.010
	Threshold	$\mathbf{0.038 \pm 0.006}$	$\mathbf{0.126 \pm 0.024}$	$\mathbf{0.047 \pm 0.006}$	$\mathbf{0.013 \pm 0.003}$
	Distribution	0.046 ± 0.007	0.131 ± 0.023	0.071 ± 0.011	0.029 ± 0.009
Crime (n=1994)	None	0.101 ± 0.008	0.363 ± 0.006	0.125 ± 0.010	0.061 ± 0.007
	Average	0.081 ± 0.016	$\mathbf{0.358 \pm 0.006}$	0.088 ± 0.016	$\mathbf{0.019 \pm 0.008}$
	Threshold	$\mathbf{0.070 \pm 0.011}$	$\mathbf{0.358 \pm 0.006}$	$\mathbf{0.073 \pm 0.017}$	0.021 ± 0.009
	Distribution	0.084 ± 0.015	0.378 ± 0.007	0.099 ± 0.015	0.027 ± 0.009

Table 5: Gaussian Forecaster Recalibration. Threshold calibration decreases the reliability gap. Despite that average calibration obtains low ECE, it can potentially increase the size of the reliability gap (see Naval dataset).

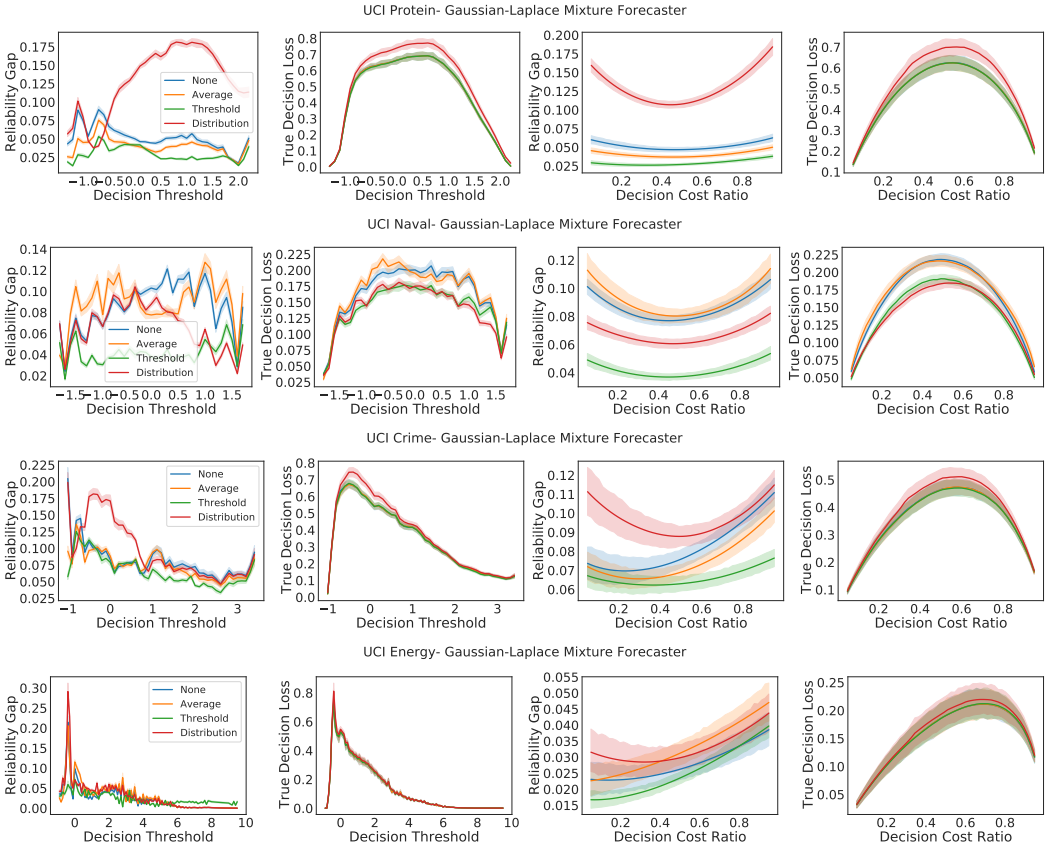

Figure 8: UCI Protein, Naval, Crime, Energy results with Gaussian-Laplace forecaster. As described in Section 5, the improvements from recalibration in the reliability gap and the decision loss are more modest because the uncalibrated forecaster that outputs Gaussian-Laplace distributions performs better than the uncalibrated forecaster that outputs Gaussian distributions. Neverthless, threshold calibration offers improvement across decision thresholds and decision costs. In contrast, distribution calibration can enlarge the reliability gap and increase the decision loss.

Dataset	Method	**Reliability Gap**	**Decision Loss**	**TCE**	**ECE**
Protein	None	0.052 ± 0.009	0.474 ± 0.008	0.054 ± 0.009	0.014 ± 0.005
(n=45730)	Average	0.041 ± 0.006	0.474 ± 0.008	0.046 ± 0.008	$\mathbf{0.005 \pm 0.001}$
	Threshold	$\mathbf{0.029 \pm 0.002}$	$\mathbf{0.473 \pm 0.009}$	$\mathbf{0.034 \pm 0.005}$	0.006 ± 0.002
	Distribution	0.130 ± 0.025	0.531 ± 0.011	0.113 ± 0.011	0.038 ± 0.007
Energy	None	0.028 ± 0.007	0.157 ± 0.014	0.075 ± 0.011	0.023 ± 0.006
(n=19735)	Average	0.029 ± 0.008	0.157 ± 0.014	0.058 ± 0.015	$\mathbf{0.008 \pm 0.004}$
	Point	$\mathbf{0.025 \pm 0.004}$	$\mathbf{0.157 \pm 0.013}$	$\mathbf{0.054 \pm 0.011}$	0.012 ± 0.004
	Distribution	0.033 ± 0.007	0.163 ± 0.015	0.089 ± 0.016	0.038 ± 0.007
Naval	None	0.086 ± 0.009	0.167 ± 0.012	0.202 ± 0.075	0.091 ± 0.039
(n=11934)	Average	0.091 ± 0.017	0.170 ± 0.012	0.083 ± 0.021	0.014 ± 0.007
	Threshold	$\mathbf{0.042 \pm 0.009}$	0.145 ± 0.011	$\mathbf{0.055 \pm 0.012}$	$\mathbf{0.013 \pm 0.007}$
	Distribution	0.067 ± 0.014	$\mathbf{0.143 \pm 0.013}$	0.183 ± 0.031	0.086 ± 0.017
Crime	None	0.081 ± 0.017	0.357 ± 0.014	0.091 ± 0.024	0.030 ± 0.010
(n=1994)	Average	0.075 ± 0.020	0.356 ± 0.015	0.079 ± 0.026	$\mathbf{0.022 \pm 0.007}$
	Threshold	$\mathbf{0.066 \pm 0.010}$	$\mathbf{0.355 \pm 0.010}$	$\mathbf{0.078 \pm 0.016}$	0.026 ± 0.007
	Distribution	0.097 ± 0.018	0.382 ± 0.021	0.093 ± 0.019	0.034 ± 0.012

Table 6: Gaussian-Laplace Mixture Forecaster Recalibration. Threshold calibration can decrease the size of the reliability gap. Distribution calibration can be challenging to enforce in model families with more parameters, and we see that it can be detrimental to the performance of the forecaster.

The Bayes decision rule with respect to the forecasted distribution is

$$\delta^*(X) = \begin{cases} 1 & \text{if } \Pr(\tilde{Y} < y_0 \mid X) \leq \frac{c_{01}-c_{11}}{c_{01}+c_{10}-c_{11}-c_{00}} \\ 0 & \text{else} \end{cases}$$

Equivalently,

$$\delta^*(X) = \begin{cases} 1 & \text{if } h[X](y_0) \leq \frac{c_{01}-c_{11}}{c_{01}+c_{10}-c_{11}-c_{00}} \\ 0 & \text{else} \end{cases}.$$

Thus, the Bayes decision rule is a threshold decision rule given by

$$\delta^*(X) = \mathbb{I}\Big(h[X](y_0) \leq \frac{c_{01}-c_{11}}{c_{01}+c_{10}-c_{11}-c_{00}}\Big).$$

B.2 Proof of Lemma 1

Proof. Let f be a forecaster $h : \mathcal{X} \to \mathcal{F}(\mathcal{Y})$ that satisfies Definition 3. Then we have that

$$\Pr[h[X](Y) \leq c] = \Pr[h[X](Y) \leq c \mid h[X](y_0) \leq 1] = c, \tag{2}$$

where $y_0 \in \mathcal{Y}$. By the law of total probability, we have that

$$\Pr[h[X](Y) \leq c] = \Pr[h[X](Y) \leq c, h[X](y_0) > \alpha] + \Pr[h[X](Y) \leq c, h[X](y_0) \leq \alpha]. \tag{3}$$

From Definition 3, we have that $\alpha \in [0,1], y_0 \in \mathcal{Y}$,

$$\begin{aligned} \Pr[h[X](y_0) \leq c \mid h[X](y_0) \leq \alpha] &= c. \\ \frac{\Pr[h[X](y_0) \leq c, h[X](y_0) \leq \alpha]}{\Pr[h[X](y_0) \leq \alpha]} &= c \\ \frac{Pr[h[X](Y) \leq c] - \Pr[h[X](Y) \leq c, h[X](y_0) > \alpha]}{1 - \Pr[h[X](y_0) > \alpha)} &= c \\ \frac{c - \Pr[h[X](Y) \leq c, h[X](y_0) > \alpha]}{1 - \Pr[h[X](y_0) > \alpha)} &= c \end{aligned}$$

Rearranging the terms, we find that

$$\Pr[h[X](Y) \leq c \mid h[X](y_0) > \alpha] = \frac{\Pr[h[X](Y) \leq c, h[X](y_0) > \alpha]}{\Pr[h[X](y_0) > \alpha]} = c.$$

Thus, if a forecaster satisfies Definition 3, then it also satisfies

$$\Pr[h[X](Y) \leq c \mid h[X](y_0) > \alpha] = c \quad \forall y_0 \in \mathcal{Y}, \forall \alpha \in [0,1], c \in [0,1].$$

□

B.3 Proof of Theorem 1

Proof of Theorem 1. Before proving the theorem we need a simple Lemma

Lemma 2. *For any pair of random variables U, V, $\mathbb{E}[U \mid V] = 0$ almost surely if and only if $\forall c \in \mathbb{R}, \mathbb{E}[U\mathbb{I}(V > c)] = 0$.*

We first show that if a forecaster h is threshold-calibrated, then the forecaster yields zero reliability gap for any threshold decision rule under any threshold loss function. Suppose we have a threshold loss function with decision threshold $y_0 \in \mathcal{Y}$.

Let $U = h^*[X](y_0)$ and $\tilde{U} = h[X](y_0)$. Suppose h satisfies threshold calibration $\Pr[h[X](Y) \leq c \mid h[X](y_0) \leq \alpha] = c$, under the new notation this implies that

$$\mathbb{E}[\mathbb{I}(h[X](Y) \leq c) - c \mid \tilde{U} \leq \alpha] = 0$$

We can further derive

$$\begin{aligned}
\mathbb{E}[U - \tilde{U} \mid \tilde{U} \leq \alpha] &= \mathbb{E}[h^*[X](y_0) - h[X](y_0) \mid \tilde{U} \leq \alpha] \\
&= \mathbb{E}[\mathbb{I}(Y \leq y_0) - h[X](y_0) \mid \tilde{U} \leq \alpha] \\
&= \mathbb{E}[\mathbb{I}(h[X](Y) \leq h[X](y_0)) - h[X](y_0) \mid \tilde{U} \leq \alpha] \qquad \text{Monotonicity} \\
&= 0
\end{aligned}$$

Therefore, we have that $\mathbb{E}[(U - \tilde{U})\mathbb{I}(\tilde{U} \leq \alpha)] = 0, \forall \alpha \in [0,1]$. We can use this fact to show that the reliability gap must be equal to 0.

For any loss function ℓ and threshold decision rule δ_h we have that the true average decision loss can be written as follows:

$$\mathbb{E}[\ell(X, Y, \delta_h(X))] = \mathbb{E}[\ell(X, Y, 1)\mathbb{I}(\delta_h(X) = 1)] + \mathbb{E}[\ell(X, Y, 0)\mathbb{I}(\delta_h(X) = 0)]$$

Similarly, the predicted average decision loss can be written as follows:

$$\mathbb{E}[\ell(X, \tilde{Y}, \delta_h(X))] = \mathbb{E}[\ell(X, Y, 1)\mathbb{I}(\delta_h(X) = 1)] + \mathbb{E}[\ell(X, Y, 0)\mathbb{I}(\delta_h(X) = 0)]$$

WLOG, it suffices to show that

$$\mathbb{E}[\ell(X, Y, 1)\mathbb{I}(\delta_h(X) = 1)] - \mathbb{E}[\ell(X, \tilde{Y}, 1)\mathbb{I}(\delta_h(X) = 1)] = 0 \quad \forall \alpha, c \in [0,1], y_0 \in \mathcal{Y}.$$

We find that

$$\begin{aligned}
&\mathbb{E}[\ell(X, Y, 1)\mathbb{I}(\delta_h(X) = 1)] - \mathbb{E}[\ell(X, \tilde{Y}, 1)\mathbb{I}(\delta_h(X) = 1)] \\
=&\mathbb{E}[c_{11}\mathbb{I}(Y \geq y_0, \delta_h(X) = 1)] + \mathbb{E}[c_{01}\mathbb{I}(Y \leq y_0, \delta_h(X) = 1)] \\
&- \mathbb{E}[c_{11}\mathbb{I}(\tilde{Y} \geq y_0, \delta_h(X) = 1)] - \mathbb{E}[c_{01}\mathbb{I}(\tilde{Y} < y_0, \delta_h(X) = 1)] \\
=&\mathbb{E}[c_{11}(1 - \mathbb{I}(Y < y_0))\mathbb{I}(\delta_h(X) = 1)] + \mathbb{E}[c_{01}\mathbb{I}(Y < y_0)\mathbb{I}(\delta_h(X) = 1)] \\
&- \mathbb{E}[c_{11}(1 - \mathbb{I}(\tilde{Y} < y_0)\mathbb{I}(\delta_h(X) = 1)] - \mathbb{E}[c_{01}\mathbb{I}(\tilde{Y} < y_0)\mathbb{I}(\delta_h(X) = 1)] \\
=&\mathbb{E}[c_{11}(1 - U)\mathbb{I}(\tilde{U} \leq \alpha)] - \mathbb{E}[c_{01}U\mathbb{I}(\tilde{U} \leq \alpha)] \\
&- \mathbb{E}[c_{11}(1 - \tilde{U})\mathbb{I}(\tilde{U} \leq \alpha)] - \mathbb{E}[c_{01}U\mathbb{I}(\tilde{U} \leq \alpha)] \\
=&0.
\end{aligned}$$

The first line follows from the definition of ℓ. The second line holds because $\mathbb{I}(Y \geq y_0) = 1 - \mathbb{I}(Y < y_0)$. The thirdline follows from the definition of U and $\tilde{U}$. The last line follows from the fact that $\mathbb{E}[U - \tilde{U} \mid \tilde{U}] = 0$ almost surely. Thus, if h is threshold calibrated, then the reliability gap is equal to zero.

We also show that the converse holds. If for any threshold loss ℓ and threshold decision rule δ_h we have

$$\mathbb{E}[\ell(X, Y, \delta_h(X))] - \mathbb{E}[\ell(X, \tilde{Y}, \delta_h(X))] = 0,$$

then $\mathbb{E}[(U - \tilde{U})\mathbb{I}(\tilde{U} \leq \alpha)] = 0$ for any $\alpha \in [0, 1]$. As a result, by Lemma 2 $\mathbb{E}[U - \tilde{U} \mid \tilde{U}] = 0$ almost surely, so we have that

$$\Pr[h[X](Y) \leq c \mid h[X](y_0) = \alpha] = c \quad \forall \alpha, c \in [0, 1], y_0 \in \mathcal{Y}$$

which is equivalent to the threshold calibration condition:

$$\Pr[h[X](Y) \leq c \mid h[X](y_0) \leq \alpha] = c \quad \forall \alpha, c \in [0, 1], y_0 \in \mathcal{Y}.$$

□

B.4 Proof of Proposition 1

Proof. Let h be a forecaster $h : \mathcal{X} \to \mathcal{F}(\mathcal{Y})$ where $\mathcal{F}(\mathcal{Y})$ is a class of continuous CDFs mapping $\mathcal{Y} \to [0, 1]$.

1. Suppose a forecaster h satisfies distribution calibration, we show that it must also be threshold-calibrated. Let $g \in \mathcal{F}(\mathcal{Y})$. For any $y_0 \in \mathcal{Y}$ and $\alpha \in [0, 1]$,

$$\begin{aligned}
\Pr[h[X](Y) \leq c \mid h[X](y_0) \leq \alpha] &= \mathbb{E}[\mathbb{I}(h[X](Y) \leq c) \mid h[X](y_0) \leq \alpha] \\
&= \mathbb{E}[\Pr[h[X](Y) \leq c \mid h[X](y_0) \leq \alpha, h[X] = g]] \\
&= \mathbb{E}[\Pr[h[X](Y) \leq c \mid g(y_0) \leq \alpha, h[X] = g]] \\
&= \mathbb{E}[\Pr[h[X](Y) \mid h[X] = g]] \\
&= c.
\end{aligned}$$

   The first line is from the definition of probability. The second is due to law of iterated expectations. The third line also follows from the fact that $h[X] = g$ determines whether $h[X](y_0) \leq \alpha$. The third line follows from the definition of distribution calibration.

2. Suppose a forecaster h satisfies threshold calibration, then it must be average-calibrated. For all $x \in \mathcal{X}, y_0 \in \mathcal{Y}, \alpha = 1$

$$\begin{aligned}
\Pr[h[X](Y) \leq c] &= \Pr[h[X](Y) \leq c \mid h[X](y_0) \leq 1] \\
&= c.
\end{aligned}$$

□

B.5 Proof of Theorem 2

Proof of Theorem 2. Denote $\mathcal{X}_0 = \{x, h^t[x](y_t) \leq \alpha_t\}$ and $\mathcal{X}_1 = \{x, h^t[x](y_t) > \alpha_t\}$ and suppose that

$$\sum_{j=0,1} P(\mathcal{X}_j) \int_y (F_{Y|\mathcal{X}_1}(y) - F^*_{Y|\mathcal{X}_j}(y))^2 dy \geq \epsilon$$

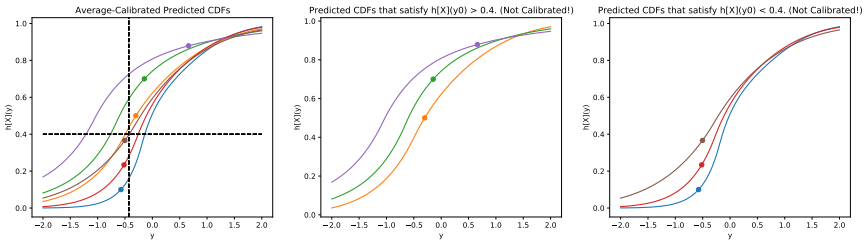

Figure 9: We partition the predicted CDFs into two groups 1) $h[X](y_0) > \alpha$, 2) $h[X](y_0) < \alpha$. The dot on each CDF denotes the location of the true label. **Left**: We observe that $h[X](Y)$ is (approximately) uniformly distributed, so the predicted CDFs satisfy average calibration. Dashed lines denote the position of y_0 and α. **Middle:** The forecaster is not average calibrated conditioned on the subset of predicted CDFs where $h[X](y_0) > \alpha$ because $h[X](Y)$ is not uniformly distributed. **Right**: The forecaster is not average calibrated conditioned on the predicted CDFs where $h[X](y_0) < \alpha$ because $h[X](Y)$ is not uniformly distributed in this part. As a result, average calibration does not imply threshold calibration.

Choose as the potential $\mathbb{E}\left[\left|\int_y F^*[X](y) - \tilde{F}_t[X](y)\right|^2\right]$, denote $\gamma_j(y) = F^*_{Y|\mathcal{X}_j}(y) - \tilde{F}_{Y|\mathcal{X}_j}(y)$ then

$$
\begin{aligned}
&\mathbb{E}\left[\int_y (F^*[X](y) - \tilde{F}_t[X](y))^2\right] - \mathbb{E}\left[\int_y (F^*[X](y) - \tilde{F}_{t+1}[X](y))^2\right] \\
&= \sum_j P(\mathcal{X}_j)\mathbb{E}\left[\int_y (F^*[X](y) - \tilde{F}_t[X](y))^2 - (F^*[X](y) - \tilde{F}_{t+1}[X](y))^2 \mid \mathcal{X}_j\right] \qquad \text{Tower} \\
&= \sum_j P(\mathcal{X}_j)\mathbb{E}\left[\int_y (F^*[X](y) - \tilde{F}_t[X](y))^2 - (F^*[X](y) - \tilde{F}_t[X](y) - \gamma_j(y))^2 \mid \mathcal{X}_j\right] \\
&= \sum_j P(\mathcal{X}_j)\mathbb{E}\left[\int_y (2F^*[X](y) - 2\tilde{F}_t[X](y) - \gamma_j(y))\gamma_j(y) \mid \mathcal{X}_j\right] \\
&= \sum_j P(\mathcal{X}_j) \int_y \gamma_j(y)^2 \geq \epsilon
\end{aligned}
$$

□

C Counterexample

In Figure 9, we give a visualization of why average calibration does not necessarily imply threshold calibration. Although the forecaster satisfies average calibration across the predicted CDFs (leftmost plot), we notice that the forecaster is not calibrated when conditioned on subsets of the predicted CDFs that satisfy $h[X](y_0) \leq \alpha$ and $h[X](y_0) > \alpha$, respectively.