
Scalable Utility-Aware Multiclass Calibration

Anonymous Author(s)

Affiliation

Address

email

Abstract

1 Ensuring that classifiers are well-calibrated, i.e., their predictions align with ob-
2 served frequencies, is a minimal and fundamental requirement for classifiers to
3 be viewed as trustworthy. Existing methods for assessing multiclass calibration
4 often focus on specific aspects associated with prediction (e.g., top-class confi-
5 dence, class-wise calibration) or utilize computationally challenging variational
6 formulations. We instead propose *utility calibration*, a general framework designed
7 to evaluate model calibration directly through the lens of downstream applications.
8 This approach measures the calibration error relative to a specific *utility function*
9 that encapsulates the goals or decision criteria relevant to the end user. As such,
10 utility calibration provides a task-specific perspective on reliability. We demon-
11 strate how this framework can *unify and re-interpret several existing calibration*
12 *metrics*, particularly allowing for more robust versions of the top-class and class-
13 wise calibration metrics, and to go beyond such binarized approaches, towards
14 assessing calibration for richer classes of downstream utilities.

1 Introduction

16 Calibration is a fundamental property of probabilistic predictors. A calibrated model produces
17 predictions that, on average, align with observed frequencies. For instance, if a weather forecaster
18 predicts a 30% chance of rain on a given day, rain should occur on approximately 30% of such days.
19 In multiclass classification problems, calibration ensures that the predicted probabilities reflect the
20 true likelihood of each class. Formally, let \mathcal{X} denote the input space, $\mathcal{Y} = \{e_1, \dots, e_C\}$ the output
21 space, where e_i is the i -th canonical basis vector in \mathbb{R}^C , and $\Delta^{C-1} := \{x \in \mathbb{R}_+^C \mid \sum_i x_i \leq 1\}$ denote
22 the simplex in \mathbb{R}^C . A predictor $f : \mathcal{X} \rightarrow \Delta^{C-1}$ is said to be perfectly calibrated with respect to
23 a distribution D over $\mathcal{X} \times \mathcal{Y}$ if $\mathbb{E}[Y \mid f(X)] = f(X)$. The most direct metric for quantifying the
24 deviation from perfect calibration is the Mean Calibration Error (MCE).

25 **Definition 1.1** (Mean Calibration Error). *For a distribution D such that $(X, Y) \sim D$ and a predic-*
26 *tor f , the mean calibration error is defined as $\text{MCE}(f) := \mathbb{E}[\|\mathbb{E}[Y \mid f(X)] - f(X)\|_2]$.*

27 Without further assumptions, the MCE is fundamentally impossible to estimate, even in the binary
28 setting [1, 2]. While assumptions like Hölder continuity of $\mathbb{E}[Y \mid f(X)]$ allow for consistent estimators
29 of $\mathbb{E}[Y \mid f(X)]$ or minimax optimal tests for $\text{MCE}(f)$ [1, 3, 4], their sample complexity scales
30 exponentially with the dimension C , making MCE estimation intractable in high dimensions.

31 Due to the difficulty of measuring MCE, multiple relaxations are proposed, falling into two main
32 categories: *binarized* and *variational*. First, binarized approaches [5–7] simplify the problem by
33 focusing on specific binary events derived from the multiclass predictions, e.g. top-class or class-wise
34 calibration. However, these methods are by nature presumptive of downstream tasks. Moreover, their
35 reliance on binning schemes or kernel estimators for the underlying binary subproblems introduce
36 sensitivity to estimator choices and can suffer from high bias [8]. Second, variational approaches
37 [9–14] assess calibration through optimization problems, such as the distance to the nearest perfectly

38 calibrated predictor or the worst-case error against a class of witness functions. Unfortunately, these
39 methods can be computationally intensive and can scale poorly as the number of classes C increases.

40 To address these limitations and provide an application-focused perspective on calibration, we
41 introduce *utility calibration*. This framework evaluates a model f by considering a downstream user
42 who employs its predictions $f(X)$. The core idea is to measure calibration error relative to a specific
43 *utility function*, denoted u , which encapsulates the goals, costs, or decision criteria relevant to this
44 end user. Utility calibration then assesses how well the *expected utility* (as estimated by the user
45 based on $f(X)$ and u) aligns with the *realized utility* (obtained when the true outcome Y is observed).
46 In practice, models often serve diverse users or a single user with multiple objectives. We thus extend
47 utility calibration to handle *classes of utility functions*. The overall utility calibration for a class \mathcal{U}
48 can be defined as the worst-case error over $u \in \mathcal{U}$, denoted $\text{UC}(f, \mathcal{U})$. A notable aspect of this
49 class-based formulation is that it provides a structured way to express and analyze various existing
50 calibration notions. In particular, by defining appropriate utility functions within \mathcal{U} , concepts such as
51 top-class and class-wise calibration can be cast within the utility calibration framework. This offers a
52 unified perspective and a superior alternative to binning for examining those notions of calibration.

53 **Contributions and outline:** In Section 2, we review related literature on calibration metrics and
54 post-hoc calibration methods. In Section 3, we define utility calibration and relate it to existing
55 measures of calibration. In addition, we demonstrate how this framework can be used to frame
56 several existing calibration concepts within a common utility-centric perspective, offering consistent
57 interpretations and providing examples of relevant utility classes. To characterize the difficulty of
58 achieving utility calibration for classes of utility functions, we introduce the notions of *proactive*
59 and *interactive* measurability. While, for rich utility classes, proactive measurability is not possible,
60 we show that interactive measurability is achievable for many classes of interest. Drawing on
61 these insights, we empirically demonstrate the application of our proposed metrics and evaluation
62 methodology, in Section 4, to that end, we formulate a practical and scalable methodology for
63 evaluating calibration against interactively measurable utility classes in Section 4.

64 **Notation:** For any vector $w \in \mathbb{R}^C$, w_i denotes its i -th component and $\gamma(w) := \arg\max_i w_i$. For a
65 probability vector $p \in \Delta^{C-1}$, we write $Z \sim p$ to denote a categorical random variable Z taking values
66 in $\mathcal{Y} = \{e_1, \dots, e_C\}$ such that $\mathbb{P}\{Z = e_i\} = p_i$, where e_i is the i -th canonical basis vector. We use
67 $\mathbb{1}\{\cdot\}$ for the indicator function. $\mathbb{E}[\cdot]$ denotes expectation, which is taken typically w.r.t. $(X, Y) \sim D$
68 and, for $k \in \mathbb{N}_+$, $[k] = \{1, \dots, k\}$. Finally, for $a, b \in \mathbb{R}$ with $a < b$, we denote $\mathbb{I}[a, b]$ to be the set
69 of closed interval subsets of $[a, b]$.

70 2 Related Work

71 In this section, we review three classical and related approaches to measuring or ensuring a form of
72 calibration, namely binarized relaxations, variational approaches, and post-hoc calibration methods.

73 First, *binarized relaxations* aim to circumvent the difficulty of measuring the calibration error of
74 a high-dimensional predictor f by measuring the MCE of a single or multiple downstream binary
75 versions of f instead. Two commonly used relaxations are the Top-Class calibration Error (TCE) [7]
76 and the Class-Wise calibration Error (CWE) [6], which are respectively defined as

$$\begin{aligned} \text{TCE}(f) &:= \mathbb{E} \left[\left| \mathbb{E}[\mathbb{1}\{Y = e_{\gamma(f(X))}\} | f(X)_{\gamma(f(X))}] - f(X)_{\gamma(f(X))} \right| \right], \\ \text{CWE}(f) &:= \sum_{i \in [C]} w_i \mathbb{E} \left[\left| \mathbb{E}[\mathbb{1}\{Y = e_i\} | f(X)_i] - f(X)_i \right| \right], \end{aligned}$$

77 where w_i is a class-dependent weight, which can be set to $1/C$, $w_i = \mathbb{P}\{Y = e_i\}$, or another
78 choice. Typically, TCE and CWE are estimated using binning schemes. Concretely, for $(B_j)_{j \in [m]}$
79 m disjoint subsets of $[0, 1]$ such that $\cup_{j \in [m]} B_j = [0, 1]$, we consider the following binned estimators

$$\text{TCE}^{\text{bin}}(f) = \sum_{j \in [m]} \left| \mathbb{E} \left[(f(X)_{\gamma(f(X))} - \mathbb{1}\{Y = e_{\gamma(f(X))}\}) \mathbb{1}\{f(X)_{\gamma(f(X))} \in B_j\} \right] \right|, \quad (2.1)$$

$$\text{CWE}^{\text{bin}}(f) = \sum_{i \in [C]} \sum_{j \in [m]} w_i \mathbb{E} \left[(f(X)_i - \mathbb{1}\{Y = e_i\}) \mathbb{1}\{f(X)_i \in B_j\} \right]. \quad (2.2)$$

80 Gupta and Ramdas [5] unified multiple instances of binarized proxies of MCE, such as TCE, CWE
81 and topK confidence calibration, introduced in [15], and proposed additional binarized reductions

which offer stronger notions of calibration. Unfortunately, the binning schemes used in such binarized proxies are known to have a large effect on the estimated error [8, 16]. Apart from the simpler equal-size bins [7] and equal-weight bins [17], multiple binning schemes built on top of different heuristics have been proposed [see, e.g., 8, 18–20]. Gupta and Ramdas [21] showed a simple equal-weight binning scheme with better sample complexity guarantees for estimating bin averages. Kumar et al. [22] developed adaptive binning schemes with guarantees for discrete f and showed that for any binning scheme, there exists a worst-case continuous f such that the bias of $\text{TCE}^{\text{bin}}(f)$ as an estimate of $\text{TCE}(f)$ is lower bounded by 0.49 (noting that by construction TCE is bounded between 0 and 1). On the other hand, there exist binning-free alternatives for binarized reductions [see, e.g., 3, 15]. Nonetheless, in an assumption-free setting, it is generally impossible to consistently estimate the MCE of binary predictors [1, 2, 23]. As such, it is generally difficult to control the calibration error defined by binarized relaxations.

Second, *variational approaches* do not strictly aim to measure the MCE. Instead, they consider alternative formulations that do not require direct estimation of the conditional expectation. For example, Distance to Calibration (DC) quantifies the calibration error of a predictor f as the distance between f and the nearest perfectly calibrated predictors [10]:

$$\text{DC}(f) := \inf_{\text{MCE}(g)=0} \mathbb{E}[\|f(X) - g(X)\|_1].$$

A unified formulation of variational measures of calibration is weighted calibration, which assesses the calibration error against a class of witness functions [9]. Concretely, let \mathcal{W} be a class of functions mapping Δ^{C-1} to $[-1, 1]^C$. Then, weighted calibration error with witness class \mathcal{W} is

$$\text{CE}_{\mathcal{W}}(f) = \sup_{w \in \mathcal{W}} \mathbb{E}_{X,Y} [\langle w(f(X)), f(X) - Y \rangle]. \quad (2.3)$$

A specific instance of weighted calibration is the Kernel Calibration Error (KCE) [24], which sets \mathcal{W} to be the unit ball of the reproducing kernel Hilbert space (RKHS) of a multivariate universal kernel. This allows for efficient computation of the supremum but it remains hard to interpret the impact of low KCE for a user of f . Błasiok et al. [10] showed that in the binary setting, $\text{DC}(f)$ and $\text{CE}_{\text{Lip}(1)}(f)$ are equivalent up to a (low-degree) polynomial scaling, where $\text{Lip}(1)$ is the class of 1-Lipschitz functions from Δ^{C-1} to $[-1, 1]$. In addition, the authors proved that, for the binary setting, $\text{CE}_{\text{Lip}(1)}(f)$ can be well approximated by the RKHS of the Laplace kernel allowing for efficient assessment of $\text{DC}(f)$ using a calibration metric originally proposed by Kumar et al. [12].

The result on the equivalence between $\text{CE}_{\text{Lip}(1)}(f)$ and $\text{DC}(f)$ was further extended to the multiclass setting in [2, Theorem 15.5.5] and [11, Lemma 3.3]. In particular, Gopalan et al. [25] showed that measuring either $\text{DC}(f)$ or $\text{CE}_{\text{Lip}(1)}(f)$ requires an exponential number of samples with respect to C [11, Theorem 3.2. and Theorem 3.4.]. Thus, even though $\text{DC}(f)$ can be efficiently assessed in the binary setting, it is quickly intractable as the dimension increases.

A particular case is *Decision calibration*, introduced by Zhao et al. [14], that tailors calibration guarantees to downstream decision-making tasks. A predictor f is considered decision calibrated of order K if, for any decision problem involving at most K actions, the expected loss computed using the model’s predictions $f(X)$ accurately matches the true expected loss incurred. Formally, for any loss function ℓ mapping an outcome-action pair to a real-valued loss, decision calibration of order K requires:

$$\mathbb{E}[\ell(\hat{Y}, \delta(f(X)))] = \mathbb{E}[\ell(Y, \delta(f(X)))],$$

where $\hat{Y} \sim f(X)$ and δ is a decision rule that picks the best action among K actions under the model’s prediction $f(X)$. This ensures that decision-makers can reliably estimate the consequences of their choices when using the predictor. A key contribution of Zhao et al. [14] is showing that decision calibration of order K can be achieved by having $\sup_{p \in P(K)} \|\mathbb{E}[(Y - f(X)) \mathbb{1}\{f(X) \in p\}]\| = 0$, where $P(K)$ is the set of polytopes with at most K supporting hyperplanes. Unfortunately, computational complexity is again an issue—Gopalan et al. [11] showed that even for $K = 2$ the computational complexity of measuring decision calibration is exponential with respect to C .

In summary, practitioners are faced with a dilemma in assessing the calibration error. On one hand, for binarized approaches, it is generally impossible to have consistent estimation of the calibration error of the binary subproblems. In addition, by preemptively only assessing specific binary subproblems, they are fundamentally presumptive of the downstream usage of the model. On the other hand,

131 variational approaches can offer more robust and well-motivated assessment of the calibration error
 132 but they are computationally infeasible as the dimension grows.

133 Independently, *post-hoc calibration* refers to techniques applied to a pre-trained model’s outputs
 134 to improve the alignment between its predicted probabilities and the true likelihood of outcomes,
 135 without altering the original model parameters. Such methods are advantageous as they decouple the
 136 calibration process from the training process.

137 Common post-hoc calibration methods often adjust the model’s outputs; popular examples include
 138 Temperature Scaling and its multi-parameter extensions, Vector Scaling and Matrix Scaling [7], which
 139 may all be regarded as a multiclass extension of Platt’s scaling [26]. Dirichlet calibration assumes the
 140 model’s predicted probability vectors can be modeled by a Dirichlet distribution, whose parameters
 141 are learned on a calibration set to transform the original probabilities [27]. Nonparametric methods
 142 such as Histogram Binning [17] and Isotonic Regression [28] learn calibration maps by discretizing
 143 the probability space or fitting monotonic (order-preserving) functions, respectively. Other methods
 144 also include: [18], which applies a specific binning strategy followed by recalibration to minimize
 145 class-wise calibration error, [29], which uses order-preserving transformations for recalibration to
 146 maintain accuracy. Finally, a related body of literature aims to improve calibration by changing or
 147 regularizing the training objective, e.g. [30, 3, 31, 12].

148 3 Utility Calibration

149 We consider the following utility-centric formulation of calibration. In particular, we are interested in
 150 the setting, where for some input X , a downstream user leverages $f(X)$ as an estimation of $\mathbb{E}[Y|X]$.
 151 Based on this estimation of the conditional expectation, the user may then take arbitrary actions or
 152 decisions. Finally, the user observes the true realization of the label Y and based on this realization,
 153 may then suffer some loss or achieve some gain. To model such a pipeline of observation, action, then
 154 consequences, we consider a utility function $u : \Delta^{C-1} \times \mathcal{Y} \rightarrow [-1, 1]$ such that $u(f(X), Y)$ models
 155 the reward obtained or the loss suffered by the decision-makers after using $f(X)$ to take arbitrary
 156 actions/decisions. In such a setting, predictability is highly desirable, in the sense that when using the
 157 predictor f , the utility obtained is similar to the utility expected. More concretely, for $\hat{Y} \sim f(X)$ and
 158 a given input X , the user can use $f(X)$ to construct the following estimate of utility:

$$v_u(X) := \mathbb{E} [u(f(X), \hat{Y})|X] = \langle f(X), \vec{u}(X) \rangle, \quad (3.1)$$

159 where $\vec{u} : \mathcal{X} \rightarrow [-1, 1]^C$ is defined as $\vec{u}(X) := (u(f(X), e_i))_{i \in C}$. Ideally, we want the function
 160 $v_u(X)$ to be an unbiased estimator of the true utility. As such, we define the utility calibration with
 161 respect to a utility function u as

$$\text{UC}(f, u) := \sup_{I \in \mathbb{I}[-1, 1]} |\mathbb{E} [(u(f(X), Y) - v_u(X)) \mathbb{1} \{v_u(X) \in I\}]| \quad (3.2)$$

162 and say that f is ε -calibrated with respect to a utility function u if $\text{UC}(f, u) \leq \varepsilon$. Note that for any
 163 $I = [a, b]$, the inner optimization problem in (3.2) can be rewritten as

$$\left| \mathbb{E} [u(f(X), Y) - u(f(X), \hat{Y}) | v_u(X) \in [a, b]] \right| \mathbb{P} \{v_u(X) \in [a, b]\}.$$

164 In words, looking at the instances where $v_u(X) \in [a, b]$, the bias between the utility the decision-
 165 maker expects to get (while using $f(X)$ to take decisions and to estimate the utility) and the actual
 166 utility the decision-maker achieves (when using $f(X)$ to take decisions), is at most ε after being
 167 weighted by the probability of the event $\{v_u(X) \in [a, b]\}$.

168 Combining (3.1) and (3.2) above, one obtains that $\text{UC}(f, u)$ is equivalent to

$$\text{UC}(f, u) = \sup_{I \in \mathbb{I}[-1, 1]} |\mathbb{E} [(Y - f(X), \vec{u}(X)) \mathbb{1} \{v_u(X) \in I\}]|. \quad (3.3)$$

169 Thus, utility calibration is equivalent to weighted calibration (2.3), with the witness class \mathcal{W} set to
 170 $\mathcal{W}(u) := \{x \mapsto \xi \vec{u}(x) \mathbb{1} \{v_u(x) \in I\} | I \in \mathbb{I}[-1, 1]\}$. In addition, our notion of utility calibration
 171 requires that the predicted label $\hat{Y} \sim f(X)$ can be used for an unbiased estimation of the utility. This
 172 is related to Outcome Indistinguishability (OI) [32], where a predictor f is considered reliable if its
 173 simulated outcomes $\hat{Y} \sim f(X)$ are computationally indistinguishable from Nature’s true outcomes
 174 Y . We also note that this perspective also connects to recent work that leverages OI variants to
 175 establish links between loss minimization guarantees, omnipredictors, and multicalibration [33–35].

3.1 Decision-Theoretic Implications of Utility Calibration

In a very recent work, for the binary classification setting, Rossellini et al. [23] introduced the CutOff calibration metric, which assesses the calibration error by measuring against the worst-case bin, and demonstrated that it provide robust decision-theoretic guarantees. We defer a more detailed discussion of CutOff calibration to Appendix B.1. By assessing the $UC(f, u)$ on the worst-case interval of $v_u(\cdot)$, our construction of utility calibration can be seen as a generalization of CutOff calibration to multiple dimensions and arbitrary utility functions, and that in fact inherits analogous decision-theoretic guarantees to the one shown in Rossellini et al. [23, Prop 2.1 and 3.2].

In particular, consider a decision rule based on thresholding the predicted utility $v_u(X)$ at some level $t_0 \in [-1, 1]$, i.e., taking the action $\hat{U}_{t_0} := \mathbb{1}\{v_u(X) \geq t_0\}$. This models the situation in which a user needs to commit a binary decision after estimating the utility using $f(X)$. Then, the quality of this decision can be assessed by the loss $\ell_{\text{util}}(\tilde{u}, \hat{U}; t) = |\tilde{u} - t| \mathbb{1}\{\hat{U} \neq \mathbb{1}\{u \geq t\}\}$, which penalizes the *deviation* between the true utility u_Y and the decision threshold t_0 when a mismatch between \hat{U}_{t_0} and the ideal decision occurs. Consequently, let $R_{\text{util}}(g; t_0) = \mathbb{E}[\ell_{\text{util}}(u(f(X), Y), \hat{U}_{t_0}; t_0)]$ be the associated risk. Then, we show that the decision process \hat{U}_{t_0} cannot significantly be improved by any simple post-processing of $v_u(\cdot)$ through a composition with a monotone function.

Proposition 3.1 (Utility Risk Gap). *Let $u : \Delta^{C-1} \times \mathcal{Y} \rightarrow [-1, 1]$ be a utility function and $v_u(X)$ be the predicted expected utility. For any threshold $t_0 \in [-1, 1]$ and the loss function ℓ_{util} as described above,*

$$R_{\text{util}}(v_u(X); t_0) - \inf_{\substack{h: [-1, 1] \rightarrow [-1, 1] \\ \text{monotone}}} R_{\text{util}}(h(v_u(X)); t_0) \leq 2UC(f, u).$$

In words, Proposition 3.1 indicates that, if f is utility calibrated, in such a binary decision-making scenario, the user can barely benefit from any monotonic post-processing to v_u . Another interpretation of $v_u(X)$ is as a regressor for the realized utility $u_Y := u(f(X), Y) \in [-1, 1]$. Similar to Rossellini et al. [23, Prop 2.1], we can show that the regressor v_u satisfies a notion of calibration itself. First, note that distance from calibration naturally extends to such a single-dimension regression problem by considering a function $g_u(X)$ to be a perfectly calibrated predictor of u_Y if $\mathbb{E}[u_Y | g_u(X)] = g_u(X)$ almost surely. We denote this extended notion of distance from calibration as $DCU(f, u)$, the Distance to Calibrated Utility Predictor for $v_u(X)$ with respect to the realized utility $u(f(X), Y)$:

$$DCU(f) := \inf_{\substack{g_u: \mathcal{X} \rightarrow [-1, 1] \\ \mathbb{E}[u_Y | g_u(X)] = g_u(X)}} \mathbb{E} |g_u(X) - v_u(X)|.$$

We show that $DCU(f, u)$ can be effectively controlled through $UC(f, u)$.

Proposition 3.2 (Utility Calibration upper Bounds DCU). *Let $u : \Delta^{C-1} \times \mathcal{Y} \rightarrow [-1, 1]$ be a utility function. Then,*

$$DCU(f) \leq \sqrt{8UC(f, u)} + UC(f, u).$$

Proposition 3.2 implies that if $UC(f, u)$ is small, then $v_u(X)$, seen as a regressor for the true utility $u(f(X), Y)$, is a calibrated predictor itself. This further strengthens the interpretation of $UC(f, u)$: not only does it *ensure actionable decisions based on $v_u(X)$* , but it also *guarantees that $v_u(X)$ itself is not far from calibration*. We thus turn to the question of how to estimate $UC(f, u)$.

3.2 Measuring $UC(f, u)$

A naturally arising question is on the difficulty of measuring and achieving a small utility calibration error. We show in Lemma 3.3 that both the computational and sample complexity of estimating $UC(f, u)$ are generally feasible and of limited dependence on the dimension, allowing its scalability to predictors with thousands of classes.

Lemma 3.3 (Estimating Utility Calibration Against a Single Function). *Let $u : \Delta^{C-1} \times \mathcal{Y} \rightarrow [-1, 1]$ be a fixed utility function and $f : \mathcal{X} \rightarrow \Delta^{C-1}$ be a given predictor. Define the empirical estimator $\widehat{UC}(f, u; S)$ based on n i.i.d. samples $S = \{(X_i, Y_i)\}_{i=1}^n \sim D^n$ as*

$$\widehat{UC}(f, u; S) := \sup_{I \in \mathbb{I}[-1, 1]} \left| \frac{1}{n} \sum_{i=1}^n [(u(f(X_i), Y_i) - V(X_i)) \mathbb{1}\{V(X_i) \in I\}] \right|.$$

216 Then, for any $\delta > 0$, with probability at least $1 - \delta$ over the draws of the sample S ,

$$|\widehat{\text{UC}}(f, u; S) - \text{UC}(f, u)| \leq \tilde{O}\left(\sqrt{\frac{\log(1/\delta)}{n}}\right). \quad (3.4)$$

217 Furthermore, $\widehat{\text{UC}}(f, u; S)$ can be computed from S in $O(n^2 + nT_{\text{eval}})$ time, where T_{eval} is the time
218 to evaluate $f(X_i)$ and $u(\cdot, \cdot)$.

219 First, we note that the constants hidden in the $\tilde{O}(\cdot)$ in (3.4) are dimension-independent. Similarly,
220 the only dimension-dependent term in the computational complexity is T_{eval} . As such, $\text{UC}(f, u)$
221 is a completely scalable notion of calibration, allowing it to be implemented for classifier with a
222 thousand classes – as exemplified in Section 4. In addition, given that $\text{UC}(f, u)$ can be formulated
223 as weighted calibration (see eq. (3.3)) and that $\widehat{\text{UC}}(f, u; S)$ is both a computationally and sample
224 efficient, we can leverage the common patching-style post-hoc calibration algorithm, eg: [9, 36, 2] to
225 recalibrate f in order to minimize $\text{UC}(f, u)$ while decreasing its Brier score. We summarize this fact
226 informally in Lemma 3.4 and defer to a more detailed discussion and experimental evaluation of the
227 recalibration patching algorithm in Appendix A.

228 **Lemma 3.4** (Informal). For $\varepsilon > 0$, there exists an algorithm, which given a classifier $f : \mathcal{X} \rightarrow \mathcal{Y}$,
229 outputs a recalibrated classifier $\tilde{f} : \mathcal{X} \rightarrow \mathcal{Y}$ such that $\text{UC}(\tilde{f}, u) \leq \varepsilon$ and its Brier score decreases:

$$\mathbb{E} \left[\|\tilde{f}(X) - Y\|_2^2 \right] \leq \mathbb{E} \left[\|f(X) - Y\|_2^2 \right].$$

230 Those encouraging facts on the utility calibration w.r.t. a single u being established, we next turn out
231 attention to Utility Calibration against a function class \mathcal{U} .

232 3.3 Utility Calibration against a Function Class

233 In many real-world scenarios, a single probabilistic predictor f might serve multiple downstream
234 users, or a single user might employ it under varying conditions or objectives. The exact utility
235 function relevant at the time of decision-making may not be known beforehand by the model provider,
236 or it might even change over time (e.g., due to changing costs, available actions, or strategic goals),
237 or might be fundamentally user-dependent.

238 Therefore, ensuring reliability often requires guarantees that hold not just for a single, pre-specified
239 utility function, but for an entire class of plausible or relevant utility functions, denoted by \mathcal{U} . This
240 provides a more robust assurance that the model’s predictions are trustworthy across a range of
241 potential downstream applications. To capture this requirement, overloading the notion, we define
242 utility calibration against a function class as the worst-case performance over the class, i.e.

$$\text{UC}(f, \mathcal{U}) = \sup_{u \in \mathcal{U}} \text{UC}(f, u). \quad (3.5)$$

243 To illustrate the practical relevance of this concept, we exhibit hereafter several examples of utility
244 classes, each motivated by different downstream tasks. We first demonstrate how to recover similar
245 notions to top-class (2.1) and class-wise (2.2) using the framework of utility calibration (3.5).

246 **Example 3.5** (Top-Class and Class-Wise Utilities ($\mathcal{U}_{\text{TCE}}, \mathcal{U}_{\text{CWE}}$)). Define the top-class utility
247 function $u_{\text{top}}(p, y) = \mathbb{1}\{y = e_{\gamma(p)}\}$, where we recall that $\gamma(p) = \arg \max_k p_k$, and the class-wise
248 utility function for class $c \in [C]$ as $u^c(p, y) = \mathbb{1}\{y = e_c\}$. The corresponding utility classes are
249 respectively $\mathcal{U}_{\text{TCE}} = \{u_{\text{top}}\}$ and $\mathcal{U}_{\text{CWE}} = \{u^c, c \in [C]\}$. It results in defining:

$$\begin{aligned} \text{UC}(f, \mathcal{U}_{\text{TCE}}) &= \sup_{I \in \mathbb{I}[0,1]} \left| \mathbb{E} \left[\left(\mathbb{1}\{Y = e_{\gamma(f(X))}\} - f(X)_{\gamma(f(X))} \right) \mathbb{1}\{f(X)_{\gamma(f(X))} \in I\} \right] \right|, \\ \text{UC}(f, \mathcal{U}_{\text{CWE}}) &= \sup_{c \in [C]} \sup_{I \in \mathbb{I}[0,1]} \left| \mathbb{E} \left[\left(\mathbb{1}\{Y = e_c\} - f(X)_c \right) \mathbb{1}\{f(X)_c \in I\} \right] \right|. \end{aligned}$$

250 In contrast to the binned estimators TCE^{bin} (2.1) and CWE^{bin} (2.2), utility calibration with the
251 classes \mathcal{U}_{TCE} and \mathcal{U}_{CWE} offers a more robust, binning-free, computable assessment. Specifically,
252 $\text{UC}(f, \mathcal{U}_{\text{TCE}})$ and $\text{UC}(f, \mathcal{U}_{\text{CWE}})$ are determined by maximizing the calibration deviation over *any*
253 possible interval $I \subseteq [0, 1]$ (and additionally over classes for \mathcal{U}_{CWE}), effectively identifying the

254 worst-case interval-based error. This approach inherently avoids fixed binning schemes, thereby cir-
 255 cumventing pathologies where bin choices drastically alter estimated errors [8, 22]. Consequently, for
 256 any binning scheme using m bins, $m \cdot \text{UC}(f, \mathcal{U}_{\text{TCE}})$ and $m \cdot \text{UC}(f, \mathcal{U}_{\text{CWE}})$ upper bound $\text{TCE}^{\text{bin}}(f)$
 257 and $\text{CWE}^{\text{bin}}(f)$ respectively, while the converse is not true. We refer to Appendix B.2 for the formal
 258 statement. Furthermore, by Proposition 3.1, a small $\text{UC}(f, \mathcal{U}_{\text{TCE}})$ guarantees that decisions based
 259 on thresholding top-class confidence are robust to monotonic recalibration, and by Proposition 3.2
 260 that this confidence is a calibrated predictor of actual top-class accuracy. Analogous guarantees hold
 261 for $\text{UC}(f, \mathcal{U}_{\text{CWE}})$ for individual class confidences, offering assurances for downstream applications.

262 Beyond the binarized perspectives offered by \mathcal{U}_{TCE} and \mathcal{U}_{CWE} , the utility calibration framework
 263 readily accommodates richer and more complex classes of utility functions. This allows us to move
 264 beyond presumptive binary events and consider more nuanced downstream applications. In particular,
 265 consider settings where the utility derived from an outcome Y is intrinsic to the outcome itself,
 266 independent of the model’s prediction $f(X)$. For example, in medical diagnosis, the cost or severity
 267 tied to a specific disease $Y = e_j$ might be a fixed value a_j , irrespective of the diagnostic prediction.
 268 Formally, such situations can be modeled using a utility function $u_a : \Delta^{C-1} \times \mathcal{Y} \rightarrow [-1, 1]$ defined
 269 by a payoff vector $a \in [-1, 1]^C$, where utility function and the expected utility are respectively
 270 $u_a(\cdot, e_j) = a_j$ and $v_{u_a}(X) = \langle f(X), a \rangle$, with a_j represents the utility if the true outcome is e_j .

271 **Example 3.6** (Linear Utilities (\mathcal{U}_{lin})). *Define the class of linear utilities as $\mathcal{U}_{\text{lin}} := \{u_a \mid a \in$
 272 $[-1, 1]^C\}$, noting that the predicted utility $v_{u_a}(X)$ is linear in the prediction $f(X)$.*

273 A small $\text{UC}(f, \mathcal{U}_{\text{lin}})$ ensures that for any payoff vector a , the predicted expected utility $v_{u_a}(X)$, as a
 274 regressor of the realized utility, is close to calibration.

275 Alternatively, in applications like information retrieval or recommender systems, the realized utility
 276 depends on the rank assigned to the true outcome $Y = e_j$. Given a model’s prediction $p =$
 277 $f(X)$, assuming p_1, \dots, p_C are distinct (or that ties are broken arbitrarily/randomly among equal
 278 coordinates), the rank of class j , denoted $\text{rank}(p, j)$, is its position across p , i.e. $\text{rank}(p, j) :=$
 279 $\sum_{i \in [C]} \mathbb{1}\{p_j \leq p_i\}$. Using a valuation vector $\theta \in [-1, 1]^C$, a rank-based utility function can then
 280 be constructed as $u_\theta(p, e_j) = \theta_{\text{rank}(p, j)}$ with the associated expected utility function $v_{u_\theta}(X) =$
 281 $\sum_{i=1}^C f(X)_i \theta_{\text{rank}(f(X), i)}$. Calibrating for such utilities ensures the model’s expected rank-based
 282 performance aligns with reality. A prominent special case is topK utility, where the valuation vector
 283 $\theta^{(K)}$ for a given $K \in [C]$ is defined such that $\theta_r^{(K)} = 1$ if $r \leq K$ and $\theta_r^{(K)} = 0$ if $r > K$.

284 **Example 3.7** (Rank-Based and Top-K Utilities ($\mathcal{U}_{\text{rank}}, \mathcal{U}_{\text{topK}}$)). *The class of general rank-based*
 285 *utilities is $\mathcal{U}_{\text{rank}} := \{u_\theta \mid \theta \in [-1, 1]^C\}$. The class of top-K utilities is then $\mathcal{U}_{\text{topK}} := \{u_{\theta^{(K)}} \mid$
 286 $K \in [C]\}$, where $\theta_r^{(K)} = \mathbb{1}\{r \leq K\}$. Equivalently, $u_K(p, e_j) = \mathbb{1}\{\text{rank}(p, j) \leq K\}$. A small
 287 $\text{UC}(f, \mathcal{U}_{\text{rank}})$ (or $\text{UC}(f, \mathcal{U}_{\text{topK}})$) ensures reliable prediction for general rank (or specifically top-K
 288 accuracy) valuations, validating the model’s ranking capabilities.*

289 As discussed in Section 2, decision calibration [14] ensures that for problems with up to K actions, the
 290 model’s predicted utility for its recommended action matches the actual realized utility. We can frame
 291 a similar guarantee within utility calibration. For any bounded loss function $l : \mathcal{Y} \times [K] \rightarrow [-1, 1]$
 292 and a prediction $p = f(X)$, the optimal action is $\delta_l(p) = \arg \min_{a \in [K]} \mathbb{E}_{\hat{Y} \sim p}[l(\hat{Y}, a)]$. The utility
 293 function is then $u_l(p, y) = -l(y, \delta_l(p))$, representing the negative loss from outcome y under action
 294 $\delta_l(p)$. The predicted expected utility is $v_{u_l}(X) = -\mathbb{E}_{\hat{Y} \sim f(X)}[l(\hat{Y}, \delta_l(f(X))) \mid X]$.

295 **Example 3.8** (Decision Calibration Utilities ($\mathcal{U}_{\text{dec}, K}$)). *Let $\mathcal{L}_K = \{l : \mathcal{Y} \times [K] \rightarrow [-1, 1]\}$ be the*
 296 *class of all bounded K -action loss functions, and the utility class is $\mathcal{U}_{\text{dec}, K} := \{u_l, l \in \mathcal{L}_K\}$. A small*
 297 *$\text{UC}(f, \mathcal{U}_{\text{dec}, K})$ implies that for any K -action decision problem $l \in \mathcal{L}_K$, the model’s prediction of*
 298 *expected utility for its chosen action $\delta_l(f(X))$ reliably reflects the achieved utility $-l(Y, \delta_l(f(X)))$.*

299 These aforementioned examples illustrate that calibrating against classes \mathcal{U} provides guarantees
 300 tailored to diverse user needs, moving beyond simplistic binarized assessments. A critical question
 301 then arises: how can $\text{UC}(f, \mathcal{U})$ be measured for a given class \mathcal{U} , which we address in the next section.

302 3.4 Measurability of utility calibration

303 Estimating $\sup_{u \in \mathcal{U}} \text{UC}(f, u)$ in (3.5) presents two key challenges: the *computational complexity* of
 304 the optimization, and the *sample complexity* required for the empirical supremum to converge to its

305 true value. We introduce the two notions of proactive and interactive measurability to decouple these
 306 two aspects.

307 **Definition 3.9** (Proactive Measurability). *The utility calibration error w.r.t. class \mathcal{U} is proactively*
 308 *measurable if there exists an algorithm A and polynomial functions $N_{\text{poly}}, T_{\text{poly}}$ such that for*
 309 *any $\varepsilon, \delta > 0$ and $n \geq N_{\text{poly}}(C, 1/\varepsilon, 1/\delta)$ samples $S \sim D^n$, algorithm $A(S)$ outputs \hat{u} satisfying*
 310 *$|\text{UC}(f, \hat{u}) - \text{UC}(f, \mathcal{U})| \leq \varepsilon$ with probability at least $1 - \delta$ and the runtime of $A(S)$ is bounded by*
 311 *$T_{\text{poly}}(C, n)$.*

312 Generally, for a finite class \mathcal{U} , if $|\mathcal{U}|$ grows polynomially in C then by Lemma 3.3 we can guarantee
 313 proactive measurability. Nonetheless, even for simple infinite classes such as \mathcal{U}_{lin} , proactive measurability
 314 reduces to a non-convex optimization problem that cannot be generally solved in polynomial
 315 time. In fact, even aiming for a weaker notion, namely *improper auditing*, Gopalan et al. [11] showed
 316 that assessing both weaker and stronger notions than $\text{UC}(f, \mathcal{U}_{\text{lin}})$ cannot be done in polynomial time
 317 in both the error ε^{-1} and the dimension C [11, Theorem 1.3, Theorem 5.2, and Theorem 8.6]. A
 318 more detailed description of Gopalan et al. [11] hardness results is in Appendix B.3. The primary
 319 bottleneck is the *computation time*. Next, we thus propose an alternative criteria of measurability that
 320 decouples the statistical guarantee from the computational complexity of verifying the supremum.

321 **Definition 3.10** (Interactive Measurability). *The utility calibration error w.r.t. class \mathcal{U} is interactively*
 322 *measurable if there exists an estimator $\widehat{\text{UC}}(f, u; S)$ and a polynomial function N_{poly} such that for*
 323 *$n \geq N_{\text{poly}}(C, 1/\varepsilon, 1/\delta)$ samples $S \sim D^n$, it holds with probability at least $1 - \delta$ that*

$$\sup_{u \in \mathcal{U}} |\widehat{\text{UC}}(f, u; S) - \text{UC}(f, u)| \leq \varepsilon.$$

324 Interactive measurability represents a much more achievable goal. For example, while decision
 325 calibration is computationally hard to measure, Zhao et al. [14] showed that it admits polynomial
 326 sample complexity. In Appendix B.4, we further demonstrate the interactive measurability of different
 327 utility classes of interest with controlled Rademacher complexity.

328 In summary, while proactively measuring the worst-case utility calibration error $\text{UC}(f, \mathcal{U}) =$
 329 $\sup_{u \in \mathcal{U}} \text{UC}(f, u)$ is often computationally prohibitive for expressive utility classes \mathcal{U} , interactive
 330 measurability allows for efficient estimation of $\text{UC}(f, u)$ uniformly for any *specific* $u \in \mathcal{U}$. Next,
 331 we leverage this distinction to propose a scalable evaluation methodology that, instead of pursuing
 332 the intractable worst-case error, characterizes the *distribution* of utility calibration errors across \mathcal{U} .
 333 This provides a more nuanced understanding of a model f 's calibration reliability over a spectrum of
 334 potential downstream applications, that we then evaluate in experiments.

335 4 Scalable Evaluation of Utility Calibration and Experiment

Scalable Evaluation of Utility Calibration. Our approach considers a probability distribution $\mathcal{D}_{\mathcal{U}}$
 over the utility class \mathcal{U} . Many utility classes of interest admit a finite-dimensional parameterization,
 making sampling from $\mathcal{D}_{\mathcal{U}}$ practical. We sample M utility functions $\{u_m\}_{m=1}^M$ from $\mathcal{D}_{\mathcal{U}}$ and, for
 each u_m , compute its estimated error $\hat{E}_{m,n} := \widehat{\text{UC}}(f, u_m; S)$ using n data points from a sample S .
 These M error estimates then form an *empirical Cumulative Distribution Function (eCDF)*,

$$\hat{F}_{E,M,n}(e) := \frac{1}{M} \sum_{m=1}^M \mathbb{1}\{\hat{E}_{m,n} \leq e\},$$

336 which serves as an empirical proxy for the true CDF, $F_E(e) := \mathbb{P}_{u \sim \mathcal{D}_{\mathcal{U}}}(\text{UC}(f, u) \leq e)$. We provide
 337 guarantees on the difference between $F_E(e)$ and $\hat{F}_{E,M,n}(e)$ in Appendix B.5.

338 In particular, \mathcal{U}_{lin} (Example 3.6) and $\mathcal{U}_{\text{rank}}$ (Example 3.7) both admit finite-dimension parameteriza-
 339 tion. For \mathcal{U}_{lin} , we construct $\mathcal{D}_{\mathcal{U}_{\text{lin}}}$ by sampling the payoff vectors a uniformly in $[-1, 1]^C$. Meanwhile,
 340 for $\mathcal{U}_{\text{rank}}$, we also sample from $\mathcal{D}_{\mathcal{U}_{\text{rank}}}$ by uniformly sampling valuation vectors $\theta \in [-1, 1]^C$, which
 341 satisfy $\theta_1 \geq \theta_2 \geq \dots \geq \theta_C$. This is to reflect a rational preference for better ranks, i.e. the higher the
 342 rank of the true realization within the predictions of $f(X)$, the higher the utility.

343 **Numerical experiments.** We now demonstrate how our approach can be used to empirically
 344 validate model calibration. For all of our experiments, we used pretrained models for ImageNet
 345 and CIFAR10/100 [37, 38]. In Appendix D, we further detail our experimental setup, provide
 346 additional results, and list the licenses of all the assets used. Here, we present the results of two

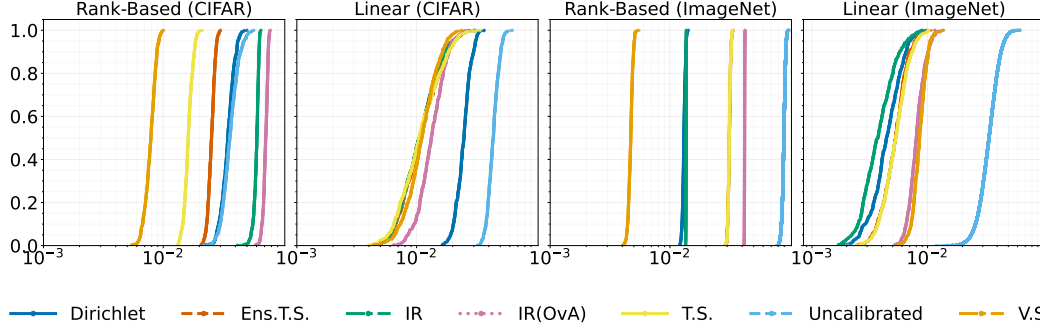


Figure 1: eCDF of utility calibration errors for ResNet20 on CIFAR100 (left two panels) and ViT on ImageNet-1K (right two panels).

settings: (1) ResNet20 [39] on CIFAR100 and a Vision Transformer ViT [40] on ImageNet-1K. For post-hoc calibration, we applied Temperature Scaling (T.S.) [26], Vector Scaling (V.S.) [41], Ensemble Temperature Scaling (Ens. T.S.) [42], and Dirichlet recalibration [27]. In addition, we fitted a shared Isotonic Regression (I.R.) [28] across different classes and an Isotonic Regression for each class using one-vs-all approach (IR OvA).

In Table 1, we present a detailed comparison for the ResNet20 model on CIFAR100. This table compares standard metrics (accuracy, Brier score), binned binarized metrics ($\text{TCE}_{\text{binned}}$, $\text{CWE}_{\text{binned}}$ with 15 equal-weight bins), and our utility calibration metrics for specific utility classes: top-class (\mathcal{U}_{TCE}), class-wise (\mathcal{U}_{CWE}), and top- K ($\mathcal{U}_{\text{topK}}$). As expected, most post-hoc methods improve Brier scores and reduce binned error over the uncalibrated model, often with minimal accuracy impact. Our binning-free utility calibration metrics, \mathcal{U}_{TCE} , \mathcal{U}_{CWE} , and $\mathcal{U}_{\text{topK}}$, show similar improvements. Notably, while \mathcal{U}_{TCE} and $\mathcal{U}_{\text{topK}}$ are equal for the uncalibrated model, they can diverge for calibrated models. Since $\mathcal{U}_{\text{topK}}$ considers all $K \in [C]$, it upper-bounds \mathcal{U}_{TCE} (the $K = 1$ case). Although calibration methods reduce \mathcal{U}_{TCE} effectively, the typically higher $\mathcal{U}_{\text{topK}}$ values can reveal miscalibration for ranks beyond top-1. This suggest $\mathcal{U}_{\text{topK}}$ as a more comprehensive benchmark.

Beyond specific utility functions, Figure 1 displays the eCDFs of utility calibration errors for broader utility classes: rank-based ($\mathcal{U}_{\text{rank}}$) and linear (\mathcal{U}_{lin}). Each eCDF, generated from $M = 1000$ sampled utility functions, shows the proportion of utilities for which the calibration error is below a certain threshold; thus, curves shifted to the left indicate superior calibration across the wider class of utility functions. For the ResNet20 on CIFAR100 (left panels), the eCDFs reveal interesting dynamics. While most post-hoc methods improved upon the uncalibrated model for \mathcal{U}_{lin} , some methods, specifically I.R. and I.R.(OvA), surprisingly worsened performance for $\mathcal{U}_{\text{rank}}$ compared to the uncalibrated model. This degradation was not apparent from the specific metrics in Table 1, underscoring the necessity of the broader perspective offered by these eCDF plots across a class of utilities. For the Vision Transformer (ViT) on ImageNet-1K (right panels), the uncalibrated model exhibits the poorest performance across both $\mathcal{U}_{\text{rank}}$ and \mathcal{U}_{lin} . Nevertheless, the eCDF plots still provide a nuanced way to compare and evaluate different post-hoc methods against each other.

In conclusion, utility calibration provides a robust, unified, and application-centric framework for evaluating classifier reliability. Its specific instantiations, \mathcal{U}_{CWE} and \mathcal{U}_{TCE} , offer superior, binning-free alternatives to traditional metrics with actionable guarantees, while $\mathcal{U}_{\text{topK}}$ presents an even more comprehensive ranking assessment. Furthermore, the eCDF plots across broader utility classes deliver crucial nuanced insights into model behavior that single-metric evaluations obscure.

Table 1: ResNet20-CIFAR100 calibration results. Comparison of post-hoc methods using Accuracy, binned ECEs ($\text{TCE}_{\text{eqBin}}$, $\text{CWE}_{\text{eqBin}}$), and utility calibration errors: \mathcal{U}_{TCE} (Top-Class), \mathcal{U}_{CWE} (Class-Wise), $\mathcal{U}_{\text{topK}}$ (Top-K). Mean \pm maximum deviation over 5 splits.

Method	Accuracy	Brier Score	$\text{CWE}_{\text{binned}}$	$\text{TCE}_{\text{binned}}$	\mathcal{U}_{CWE}	\mathcal{U}_{TCE}	$\mathcal{U}_{\text{topK}}$
Uncalibrated	0.677 \pm 0.010	0.480 \pm 0.015	0.00214 \pm 0.00016	0.1600 \pm 0.008	0.0124 \pm 0.0011	0.1590 \pm 0.015	0.1590 \pm 0.015
Dirichlet	0.666 \pm 0.010	0.457 \pm 0.008	0.00194 \pm 0.00014	0.0727 \pm 0.0160	0.0111 \pm 0.0004	0.0709 \pm 0.0165	0.0818 \pm 0.0154
IR	0.677 \pm 0.010	0.444 \pm 0.011	0.00186 \pm 0.00006	0.0264 \pm 0.0033	0.0113 \pm 0.0005	0.0310 \pm 0.0071	0.0756 \pm 0.0086
IR (OvA)	0.674 \pm 0.010	0.454 \pm 0.011	0.00156 \pm 0.00016	0.0454 \pm 0.0103	0.0108 \pm 0.0011	0.0467 \pm 0.0190	0.0927 \pm 0.0091
T.S.	0.677 \pm 0.010	0.440 \pm 0.014	0.00188 \pm 0.00008	0.0250 \pm 0.0066	0.0114 \pm 0.0005	0.0322 \pm 0.0090	0.0367 \pm 0.0046
Ens.T.S.	0.677 \pm 0.010	0.440 \pm 0.010	0.00196 \pm 0.00006	0.0212 \pm 0.0045	0.0114 \pm 0.0005	0.0304 \pm 0.0063	0.0393 \pm 0.0056
V.S.	0.680 \pm 0.010	0.435 \pm 0.010	0.00150 \pm 0.00010	0.0334 \pm 0.0117	0.0107 \pm 0.0010	0.0375 \pm 0.0148	0.0403 \pm 0.0121

References

- [1] Donghwan Lee, Xinmeng Huang, Hamed Hassani, and Edgar Dobriban. T-cal: An optimal test for the calibration of predictive models. *Journal of Machine Learning Research*, 24(335):1–72, 2023.
- [2] John C. Duchi. Information theory and statistics. <https://web.stanford.edu/class/stats311/lecture-notes.pdf>, 2024. Lecture Notes for STATS 311 / EE 377, Stanford University. Version from March 12, 2024. Accessed: April 30, 2025.
- [3] Teodora Popordanoska, Raphael Sayer, and Matthew Blaschko. A consistent and differentiable lp canonical calibration error estimator. *Advances in Neural Information Processing Systems*, 35:7933–7946, 2022.
- [4] Alexandre B Tsybakov. Nonparametric estimators. *Introduction to Nonparametric Estimation*, pages 1–76, 2009.
- [5] Chirag Gupta and Aaditya Ramdas. Top-label calibration and multiclass-to-binary reductions. In *International Conference on Learning Representations*, 2022.
- [6] Michael Panchenko, Anes Benmerzoug, and Miguel de Benito Delgado. Class-wise and reduced calibration methods. In *2022 21st IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 1093–1100. IEEE, 2022.
- [7] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330. PMLR, 2017.
- [8] Rebecca Roelofs, Nicholas Cain, Jonathon Shlens, and Michael C Mozer. Mitigating bias in calibration error estimation. In *International Conference on Artificial Intelligence and Statistics*, pages 4036–4054. PMLR, 2022.
- [9] Christopher Jung, Changhwa Lee, Mallesh Pai, Aaron Roth, and Rakesh Vohra. Moment multicalibration for uncertainty estimation. In *Conference on Learning Theory*, pages 2634–2678. PMLR, 2021.
- [10] Jarosław Błasiok, Parikshit Gopalan, Lunjia Hu, and Preetum Nakkiran. A unifying theory of distance from calibration. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, pages 1727–1740, 2023.
- [11] Parikshit Gopalan, Lunjia Hu, and Guy N Rothblum. On computationally efficient multi-class calibration. *arXiv preprint arXiv:2402.07821*, 2024.
- [12] Aviral Kumar, Sunita Sarawagi, and Ujjwal Jain. Trainable calibration measures for neural networks from kernel mean embeddings. In *International Conference on Machine Learning*, pages 2805–2814. PMLR, 2018.
- [13] David Widmann, Fredrik Lindsten, and Dave Zachariah. Calibration tests in multi-class classification: A unifying framework. *Advances in neural information processing systems*, 32, 2019.
- [14] Shengjia Zhao, Michael Kim, Roshni Sahoo, Tengyu Ma, and Stefano Ermon. Calibrating predictions to decisions: A novel approach to multi-class calibration. *Advances in Neural Information Processing Systems*, 34:22313–22324, 2021.
- [15] Kartik Gupta, Amir Rahimi, Thalaiyasingam Ajanthan, Thomas Mensink, Cristian Sminchisescu, and Richard Hartley. Calibration of neural networks using splines. In *International Conference on Learning Representations*, 2021. URL <https://openreview.net/forum?id=eQe8DEWNN2W>.
- [16] Sebastian Gruber and Florian Buettner. Better uncertainty calibration via proper scores for classification and beyond. *Advances in Neural Information Processing Systems*, 35:8618–8632, 2022.

- [17] Bianca Zadrozny and Charles Elkan. Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. In *Icml*, volume 1, pages 609–616, 2001.
- [18] Kanil Patel, William Beluch, Bin Yang, Michael Pfeiffer, and Dan Zhang. Multi-class uncertainty calibration via mutual information maximization-based binning. *arXiv preprint arXiv:2006.13092*, 2020.
- [19] Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 29, 2015.
- [20] Jeremy Nixon, Michael W Dusenberry, Linchuan Zhang, Ghassen Jerfel, and Dustin Tran. Measuring calibration in deep learning.
- [21] Chirag Gupta and Aaditya Ramdas. Distribution-free calibration guarantees for histogram binning without sample splitting. In *International conference on machine learning*, pages 3942–3952. PMLR, 2021.
- [22] Ananya Kumar, Percy S Liang, and Tengyu Ma. Verified uncertainty calibration. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/f8c0c968632845cd133308b1a494967f-Paper.pdf.
- [23] Raphael Rossellini, Jake A Soloff, Rina Foygel Barber, Zhimei Ren, and Rebecca Willett. Can a calibration metric be both testable and actionable? *arXiv preprint arXiv:2502.19851*, 2025.
- [24] Zhen Lin, Shubhendu Trivedi, and Jimeng Sun. Taking a step back with KCal: Multi-class kernel-based calibration for deep neural networks. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=p_jly5QFB7.
- [25] Parikshit Gopalan, Michael P Kim, Mihir A Singhal, and Shengjia Zhao. Low-degree multicalibration. In *Conference on Learning Theory*, pages 3193–3234. PMLR, 2022.
- [26] 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.
- [27] Meelis Kull, Miquel Perello Nieto, Markus Kängsepp, Telmo Silva Filho, Hao Song, and Peter Flach. Beyond temperature scaling: Obtaining well-calibrated multi-class probabilities with dirichlet calibration. *Advances in neural information processing systems*, 32, 2019.
- [28] Bianca Zadrozny and Charles Elkan. Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 694–699, 2002.
- [29] Amir Rahimi, Amirreza Shaban, Ching-An Cheng, Richard Hartley, and Byron Boots. Intra order-preserving functions for calibration of multi-class neural networks. *Advances in Neural Information Processing Systems*, 33:13456–13467, 2020.
- [30] Jishnu Mukhoti, Viveka Kulharia, Amartya Sanyal, Stuart Golodetz, Philip Torr, and Puneet Dokania. Calibrating deep neural networks using focal loss. *Advances in neural information processing systems*, 33:15288–15299, 2020.
- [31] Charlie Marx, Sofian Zalouk, and Stefano Ermon. Calibration by distribution matching: Trainable kernel calibration metrics. *Advances in Neural Information Processing Systems*, 36, 2024.
- [32] Cynthia Dwork, Michael P. Kim, Omer Reingold, Guy N. Rothblum, and Gal Yona. Outcome indistinguishability. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2021, page 1095–1108, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450380539. doi: 10.1145/3406325.3451064. URL <https://doi.org/10.1145/3406325.3451064>.

- [33] Parikshit Gopalan, Adam Tauman Kalai, Omer Reingold, Vatsal Sharan, and Udi Wieder. Omnipredictors. In *13th Innovations in Theoretical Computer Science Conference (ITCS 2022)*, pages 79–1. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2022.
- [34] Parikshit Gopalan, Lunjia Hu, Michael P Kim, Omer Reingold, and Udi Wieder. Loss minimization through the lens of outcome indistinguishability. In *14th Innovations in Theoretical Computer Science Conference (ITCS 2023)*, pages 60–1. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2023.
- [35] Parikshit Gopalan, Michael Kim, and Omer Reingold. Swap agnostic learning, or characterizing omniprediction via multicalibration. *Advances in Neural Information Processing Systems*, 36: 39936–39956, 2023.
- [36] Ursula Hébert-Johnson, Michael Kim, Omer Reingold, and Guy Rothblum. Multicalibration: Calibration for the (computationally-identifiable) masses. In *International Conference on Machine Learning*, pages 1939–1948. PMLR, 2018.
- [37] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.
- [38] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images, 2009.
- [39] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [40] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL <https://openreview.net/forum?id=YicbFdNTTy>.
- [41] Meelis Kull, Telmo M Silva Filho, and Peter Flach. Beyond sigmoids: How to obtain well-calibrated probabilities from binary classifiers with beta calibration. *Electronic Journal of Statistics*, 11:5052–5080, 2017.
- [42] Jize Zhang, Bhavya Kailkhura, and T Yong-Jin Han. Mix-n-match: Ensemble and compositional methods for uncertainty calibration in deep learning. In *International conference on machine learning*, pages 11117–11128. PMLR, 2020.
- [43] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. Foundations of machine learning. adaptive computation and machine learning, 2018.
- [44] Andreas Maurer. A vector-contraction inequality for rademacher complexities. In *Algorithmic Learning Theory: 27th International Conference, ALT 2016, Bari, Italy, October 19-21, 2016, Proceedings 27*, pages 3–17. Springer, 2016.
- [45] Andrey Kupavskii. The vc-dimension of k-vertex d-polytopes. *Combinatorica*, 40(6):869–874, 2020.
- [46] Pascal Massart. The tight constant in the dvoretzky-kiefer-wolfowitz inequality. *The annals of Probability*, pages 1269–1283, 1990.
- [47] Francis Bach. *Learning theory from first principles*. MIT press, 2024.
- [48] Yaofo Chen. pytorch-cifar-models: Pretrained models for CIFAR10/100 in PyTorch, February 2020. URL <https://github.com/chenyaofo/pytorch-cifar-models>.
- [49] Ross Wightman. PyTorch Image Models. URL <https://github.com/huggingface/pytorch-image-models>.

- 518 [50] Tiago Salvador. Calibration baselines. [https://github.com/tiagosalvador/](https://github.com/tiagosalvador/calibration-baselines)
519 [calibration-baselines](https://github.com/tiagosalvador/calibration-baselines), 8 2022. URL [https://github.com/tiagosalvador/](https://github.com/tiagosalvador/calibration-baselines)
520 [calibration-baselines](https://github.com/tiagosalvador/calibration-baselines). Last commit August 2022. Accessed: May 22, 2025.
- 521 [51] Tim Head, MechCoder, Gilles Louppe, Iaroslav Shcherbatyi, fcharras, Zé Vinícius, cmmalone,
522 Christopher Schröder, nel215, Nuno Campos, Todd Young, Stefano Cereda, Thomas Fan,
523 rene rex, Kejia (KJ) Shi, Justus Schwabedal, carlosdanielcsantos, Hvass-Labs, Mikhail Pak,
524 SoManyUsernamesTaken, Fred Callaway, Loïc Estève, Lilian Besson, Mehdi Cherti, Karlson
525 Pfannschmidt, Fabian Linzberger, Christophe Cauet, Anna Gut, Andreas Mueller, and Alexander
526 Fabisch. scikit-optimize/scikit-optimize: v0.5.2, March 2018. URL [https://doi.org/10.](https://doi.org/10.5281/zenodo.1207017)
527 [5281/zenodo.1207017](https://doi.org/10.5281/zenodo.1207017).
- 528 [52] Dutch Hansen, Siddhartha Devic, Preetum Nakkiran, and Vatsal Sharan. When is multicalibration
529 post-processing necessary? In *The Thirty-eighth Annual Conference on Neural Information*
530 *Processing Systems*.
- 531 [53] Xiaohan Ding, Xiangyu Zhang, Ningning Ma, Jungong Han, Guiguang Ding, and Jian Sun.
532 Repvgg: Making vgg-style convnets great again. In *Proceedings of the IEEE/CVF conference*
533 *on computer vision and pattern recognition*, pages 13733–13742, 2021.
- 534 [54] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines
535 for efficient cnn architecture design. In *Proceedings of the European conference on computer*
536 *vision (ECCV)*, pages 116–131, 2018.
- 537 [55] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural
538 networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019.

NeurIPS Paper Checklist

IMPORTANT, please:

- Delete this instruction block, but keep the section heading “NeurIPS Paper Checklist”,
- Keep the checklist subsection headings, questions/answers and guidelines below.
- Do not modify the questions and only use the provided macros for your answers.

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper’s contributions and scope?

Answer: [Yes]

Justification: The main aim of the paper is to present a unified framework for assessing calibration that allows for recovering similar notions to existing metrics while circumventing some the difficulties/limitations in assessing them. In addition, it allows going beyond binarized reductions and developing scalable assessment against infinite class through CDF curves. We present the framework, cite the literature to highlight the limitations of existing approaches, and provide proofs in the Appendix.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [NA]

Justification: The paper aims to introduce a new perspective on evaluating calibration. There are many limitations related to calibration: it is an easy notion to satisfy by trivial predictors, it is hard to measure, it is not the strongest guarantee for trustworthy deployment of machine learning models. Nonetheless, we believe that those limitations are inherit to the underlying problem rather than to the paper itself. Other aspects of the paper can be seen as limitations. For example, proactive measurability is hard, so we propose assessing infinite classes through a distributional approach. We find it hard to judge whether the hardness of proactive measurability is in itself a limitation or not, making this question hard to answer.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.

- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: Proofs included in the Appendix.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We used pretrained models for reproducibility. More detailed description of the experimental setup and additional results are available in the appendix.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.

- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Code and detailed instructions are available in the supplemental material. We intend to open source the code upon acceptance.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We used pretrained accessible models and standard datasets. Additional hyperparameter are further specified in the appendix.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.

- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We report results over multiple splits using the mean and the maximum deviation from it. We also include standard deviation in the Appendix.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: Computational resources used are detailed in Appendix D.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Justification: We satisfy NeurIPS Code of Ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.

- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [NA]

Justification: This work applies to general classifiers and is not specifically tied to particular applications.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: The paper poses no such risk.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We cited the original assets. The licenses of the assets used are detailed in Appendix D.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. New assets

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: We use standard datasets and accessible open-source pretrained models. Other aspects of the experiments are implemented in code and included in the supplemental material.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. Crowdsourcing and research with human subjects

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: No crowdsourcing or research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. Institutional review board (IRB) approvals or equivalent for research with human subjects

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: No crowdsourcing or research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. Declaration of LLM usage

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The paper does not tackle LLMs.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.

881	Contents	
882	1 Introduction	1
883	2 Related Work	2
884	3 Utility Calibration	4
885	3.1 Decision-Theoretic Implications of Utility Calibration	5
886	3.2 Measuring $UC(f, u)$	5
887	3.3 Utility Calibration against a Function Class	6
888	3.4 Measurability of utility calibration	7
889	4 Scalable Evaluation of Utility Calibration and Experiment	8
890	A Post-Hoc Patching Algorithm for Utility Calibration	22
891	B Deferred Content	23
892	B.1 CutOff Calibration	23
893	B.2 Bounding Binned Estimators using Utility Calibration	24
894	B.3 Hardness of Proactive Measurability	24
895	B.4 Interactive Measurability	25
896	B.5 Quantile Guarantees	28
897	C Proofs	29
898	C.1 Proof of Proposition 3.1	30
899	C.2 Proof of Proposition 3.2	30
900	C.3 Proof of Lemma 3.3	32
901	D Additional Experiments	32

Organization of the Appendix

The appendix provides supplementary material, organized as follows: Appendix A presents a post-hoc patching algorithm for utility calibration. Appendix B contains deferred discussions on CutOff calibration (Appendix B.1), binned estimators (Appendix B.2), proactive measurability hardness (Appendix B.3), interactive measurability (Appendix B.4), and guarantees for the plotted eCDF curves (Appendix B.5). Appendix C includes proofs of statements introduced in the main text. Finally, Appendix D details our experimental setup and provides additional results.

A Post-Hoc Patching Algorithm for Utility Calibration

This section details a post-hoc calibration algorithm aimed at reducing the utility calibration error $UC(f, \mathcal{U})$ with respect for a specific utility function class \mathcal{U} . As established in the main text (eq. (3.3)), utility calibration can be cast as a form of weighted calibration. Recall that for a single u ,

$$UC(f, u) = \sup_{w \in \mathcal{W}(u)} \mathbb{E}[\langle f(X) - Y, w(f(X)) \rangle],$$

where $\mathcal{W}(u) = \{x \rightarrow \xi \vec{u}(X) \mathbb{1}\{v_u(X) \in I\} \mid I \in \mathbb{I}[-1, 1], \xi \in \{-1, 1\}\}$. This naturally extends to a utility function class \mathcal{U} by defining $\mathcal{W}(\mathcal{U}) = \cup_{u \in \mathcal{U}} \mathcal{W}(u)$. Then it holds that $UC(f, \mathcal{U}) = CE_{\mathcal{W}(\mathcal{U})}(f)$.

Given that utility calibration can be cast as weighted calibration, algorithms from the literature can be used to post-hoc calibrate f such that it has a small utility calibration error. In particular, Algorithm 1 iteratively finds the *worst* witness $w \in \mathcal{W}(\mathcal{U})$ and adjusts the predictor f to correct for this specific violation. This approach is common in (multi)calibration literature, for example [36, 25, 9], a more general version is presented in [2, Chapter 15]. A key property is that these adjustments can be made such that the model’s Brier score decreases in every iteration.

Algorithm 1 Iterative Patching for Utility Calibration

```

1: Input: Initial predictor  $f^{(0)} : \mathcal{X} \rightarrow \Delta^{C-1}$ , witness class  $\mathcal{W}(\mathcal{U})$ , target tolerance  $\varepsilon > 0$ . Set
    $t \leftarrow 0$ 
2: loop
3:   Find  $w_t \in \operatorname{argmax}_{w \in \mathcal{W}(\mathcal{U})} \mathbb{E}[\langle f^{(t)}(X) - Y, w(f^{(t)}(X)) \rangle]$ .
4:   Let  $\text{err}_t \leftarrow \mathbb{E}[\langle f^{(t)}(X) - Y, w_t(f^{(t)}(X)) \rangle]$ .
5:   if  $\text{err}_t \leq \varepsilon$  then
6:     break
7:   end if
8:   Choose stepsize  $\eta_t$ .
9:   Update predictor:  $f^{(t+1)}(X) \leftarrow \pi_{\Delta^{C-1}}(f^{(t)}(X) - \eta_t w_t(f^{(t)}(X)))$ .
10:   $t \leftarrow t + 1$ .
11: end loop
12: Return  $f^{(t)}$ .

```

Here $\pi_{\Delta^{C-1}}$ is the projection onto the simplex and η_t is a, possibly adaptive, stepsize.

Proposition A.1 (Convergence and Brier Score Guarantee). *Assume oracle access to compute err_t and the corresponding witness $w_t \in \mathcal{W}(\mathcal{U})$ at each iteration t . Let C be the number of classes. With the stepsize $\eta_t = \text{err}_t/C$, Algorithm 1 terminates in $T = O(C/\varepsilon^2)$ iterations*

Proof. We include the proof for completeness. It follows the approach of [36] in using the Brier score as a potential function and showing that it monotonically decreases across iterations. It is the same as the version in Gopalan et al. [11]. A more general version can be found in Duchi [2, Chapter

15.]. The change in Brier score $L(f) = \mathbb{E}[\|Y - f(X)\|_2^2]$ from $f^{(t)}$ to $f^{(t+1)}$ is:

$$\begin{aligned} L(f^{(t+1)}) - L(f^{(t)}) &\leq \mathbb{E} \left[\|Y - (f^{(t)}(X) - \eta_t w_t(f^{(t)}(X)))\|_2^2 - \|Y - f^{(t)}(X)\|_2^2 \right] \\ &= \mathbb{E} \left[2\eta_t \left\langle Y - f^{(t)}(X), w_t(f^{(t)}(X)) \right\rangle + \eta_t^2 \|w_t(f^{(t)}(X))\|_2^2 \right] \\ &= \mathbb{E} \left[-2\eta_t \left\langle f^{(t)}(X) - Y, w_t(f^{(t)}(X)) \right\rangle + \eta_t^2 \|w_t(f^{(t)}(X))\|_2^2 \right] \\ &= -2\eta_t \text{err}_t + \eta_t^2 \mathbb{E}[\|w_t(f^{(t)}(X))\|_2^2]. \end{aligned}$$

Each witness $w \in \mathcal{W}(\mathcal{U})$ is of the form $p \mapsto \xi \vec{u}(p) \mathbb{1}\{v_u(p) \in I\}$ for $\xi \in \{-1, 1\}$ and $u \in \mathcal{U}$. Since $u(\cdot, e_j) \in [-1, 1]$, $\|\vec{u}(p)\|_2^2 \leq \|\vec{u}(p)\|_\infty \|\vec{u}(p)\|_1 \leq C$. Thus, $\|w_t(p)\|_2^2 \leq C$. Substituting $\eta_t = \text{err}_t/C$:

$$\begin{aligned} L(f^{(t+1)}) - L(f^{(t)}) &\leq -2\frac{\text{err}_t}{C} \text{err}_t + \left(\frac{\text{err}_t}{C}\right)^2 \mathbb{E}[\|w_t(f^{(t)}(X))\|_2^2] \\ &\leq -\frac{2\text{err}_t^2}{C} + \frac{\text{err}_t^2 C}{C^2} = -\frac{\text{err}_t^2}{C}. \end{aligned}$$

This proves that the Brier score does not increase and strictly decreases if $\text{err}_t > 0$. If the algorithm continues, it is because $\text{err}_t > \varepsilon$, so the decrease is at least ε^2/C . Since $L(f) \in [0, 2]$ (as $\|Y - f(X)\|_2^2 \leq \|Y\|_2^2 + \|f(X)\|_2^2 \leq 1 + 1 = 2$), and each step where $\text{err}_t > \varepsilon$ decreases $L(f)$ by at least ε^2/C , the algorithm must terminate in at most $O(C/\varepsilon^2)$ such steps. \square

Empirical Implementation and Sample Complexity. In practice, direct computation of expectations within Algorithm 1 is infeasible. Instead, the algorithm is implemented using a dataset S^t of N i.i.d. samples at each iteration t . Both the maximization step to find the witness w_t and the computation of the error err_t are performed using empirical averages $\hat{\mathbb{E}}_N[\cdot]$ over S^t .

The theoretical convergence guarantees of Proposition A.1 (i.e., termination in $O(C/\varepsilon^2)$ iterations) can be extended to this empirical setting, provided that the empirically estimated error $\widehat{\text{err}}_t := \hat{\mathbb{E}}_N[\langle f^{(t)}(X) - Y, w_t(X) \rangle]$ is a sufficiently accurate approximation of the true error err_t . Specifically, if $\widehat{\text{err}}_t$ is within $O(\varepsilon)$ of err_t whenever $\text{err}_t > \varepsilon$, the iteration complexity remains $O(C/\varepsilon^2)$.

The number of samples N required per iteration to achieve an $O(\varepsilon)$ -accurate estimation of err_t , with probability $1 - \delta$, depends on the complexity of the witness class $\mathcal{W}(\mathcal{U})$. For the Top-Class utility \mathcal{U}_{TCE} , where $|\mathcal{U}| = 1$, using Lemma 3.3, $N \leq \tilde{O}\left(\frac{\log(1/\delta)}{\varepsilon^2}\right)$. Similarly, Class-Wise utility \mathcal{U}_{CWE} and Top-K utility $\mathcal{U}_{\text{topK}}$, where for both $|\mathcal{U}| = C$, using a simple union bound, we recover $N \leq \tilde{O}\left(\frac{\log(C/\delta)}{\varepsilon^2}\right)$.

B Deferred Content

B.1 CutOff Calibration

In Section 3, we highlighted that our utility calibration framework, particularly its focus on worst-case interval-based deviations of predicted utility, can be seen as a natural extension of the binary CutOff calibration concept to multiclass scenarios and general utility functions. This extension preserves important decision-theoretic properties. For the binary setting ($Y \in \{0, 1\}$, $f : \mathcal{X} \rightarrow [0, 1]$), Rossellini et al. [23] demonstrate that if the metric

$$\Delta_{\text{Cutoff}}(f) := \sup_{I \in \mathbb{I}[0,1]} |\mathbb{E}[(Y - f(X)) \mathbb{1}\{f(X) \in I\}]|$$

is small, then a simple decision rule $\hat{Y}_\tau : X \rightarrow \{0, 1\}$ of the form $\hat{Y}_\tau = \mathbb{1}\{f(X) \geq \tau\}$ evaluated against its associated binary decision loss, cannot be substantially improved by monotonic post-hoc calibration. More concretely, let $R_{\text{bd}}(g; \tau) := \mathbb{E}[\ell_{\text{bd}}(Y, \hat{Y}_\tau; \tau)]$ be the risk under the binary decision loss $\ell_{\text{bd}}(Y, \hat{Y}; \tau) = \tau(1 - Y)\hat{Y} + (1 - \tau)Y(1 - \hat{Y})$. Then, Rossellini et al. [23, Prop. 3.2] show that for any $\tau \in [0, 1]$:

$$R_{\text{bd}}(f; \tau) - \inf_{\substack{h: [0,1] \rightarrow [0,1] \\ \text{monotone}}} R_{\text{bd}}(h \circ f; \tau) \leq 2\Delta_{\text{Cutoff}}(f). \quad (\text{B.1})$$

962 This guarantee implies that if $\Delta_{\text{Cutoff}}(f)$ is small, the decision-maker, thresholding $f(X)$ to make
 963 binary decision, gains little by applying any monotonic recalibration to $f(X)$. Similarly, they showed
 964 that in the binary case, CutOff calibration error can be used to bound distance from calibration. As
 965 such, Propositions 3.1 and 3.2 extend the results of Rossellini et al. [23] to the multiclass setting.

966 B.2 Bounding Binned Estimators using Utility Calibration

967 To illustrate the relationship between binned estimations of calibration error and utility calibration,
 968 let $p_X := f(X)_{\gamma(f(X))}$ and correctness indicator $Y_X := \mathbb{1}\{Y = e_{\gamma(f(X))}\}$, definitions for m bins
 969 $(B_j)_{j \in [m]}$ are:

$$\begin{aligned} \text{TCE}^{\text{bin}}(f) &= \sum_{j=1}^m |\mathbb{E}[(p_X - Y_X) \mathbb{1}\{p_X \in B_j\}]|, \\ \text{UC}(f, \mathcal{U}_{\text{TCE}}) &= \sup_{I \in \mathbb{I}[0,1]} |\mathbb{E}[(Y_X - p_X) \mathbb{1}\{p_X \in I\}]|. \end{aligned}$$

970 Each binned term $|\mathbb{E}[(p_X - Y_X) \mathbb{1}\{p_X \in B_j\}]| \leq \text{UC}(f, \mathcal{U}_{\text{TCE}})$ (by setting $I = B_j$), thus
 971 $\text{TCE}^{\text{bin}}(f) \leq m \cdot \text{UC}(f, \mathcal{U}_{\text{TCE}})$. Conversely, small binned errors do not imply small utility cal-
 972 ibration, as binned errors can cancel within bins, while utility calibration is the supremum over
 973 intervals.

974 For instance, let p_X be 0.45 or 0.55 (each with probability 0.5), with $\mathbb{E}[Y_X | p_X = 0.45] = 0.05$ and
 975 $\mathbb{E}[Y_X | p_X = 0.55] = 0.95$. If $\text{TCE}^{\text{bin}}(f)$ uses bins $B_1 = [0, 1/3)$, $B_2 = [1/3, 2/3)$, $B_3 = [2/3, 1]$,
 976 the expected error is 0. Thus, $\text{TCE}^{\text{bin}}(f) = 0$.

977 However, the inverse is not true. For example, for $\text{UC}(f, \mathcal{U}_{\text{TCE}})$, consider
 978 the interval $I_1 = [0.45, 0.46]$. The term $|\mathbb{E}[(Y_X - p_X) \mathbb{1}\{p_X \in I_1\}]|$ becomes
 979 $|P(p_X = 0.45) \cdot (\mathbb{E}[Y_X | p_X = 0.45] - 0.45)| = 0.2$. Since $\text{UC}(f, \mathcal{U}_{\text{TCE}})$ is the supremum
 980 over such intervals, $\text{UC}(f, \mathcal{U}_{\text{TCE}}) \geq 0.2$. The binned estimator indicates perfect calibration, while
 981 $\text{UC}(f, \mathcal{U}_{\text{TCE}})$ does not.

982 B.3 Hardness of Proactive Measurability

Proactive measurability for a utility class \mathcal{U} , as defined in Definition 3.9, necessitates an algorithm
 to efficiently find $\hat{u} \in \mathcal{U}$, whose utility calibration error $\text{UC}(f, \hat{u})$ approximates $\sup_{u \in \mathcal{U}} \text{UC}(f, u)$.
 This is equivalent to efficiently finding an approximate worst-case function from the witness class

$$\mathcal{W}(\mathcal{U}) = \bigcup_{u \in \mathcal{U}} \{X \mapsto \xi \vec{u}(X) \mathbb{1}\{v_u(X) \in I\} \mid I \in \mathbb{I}[-1, 1], \xi \in \{-1, 1\}\},$$

983 given that the worst-case interval for a fixed u is efficiently findable (Lemma 3.3). The work of
 984 Gopalan et al. [11] establishes computational hardness for “auditing with a witness” for related,
 985 expressive classes of witness functions.

986 **Definition B.1** (Auditing with a witness [11]). *An (α, β) auditor for a witness class $\mathcal{W}_{\text{target}}$ is an*
 987 *algorithm that, when given access to a distribution \mathcal{D} where $\text{CE}_{\mathcal{W}_{\text{target}}}(\mathcal{D}) > \alpha$, returns any function*
 988 *$w' : \Delta^{C-1} \rightarrow [-1, 1]^C$ such that $\mathbb{E}_{(X, Y) \sim \mathcal{D}}[\langle Y - f(X), w'(f(X)) \rangle] \geq \beta$.*

989 First, auditing with a witness is an *easier task than proactive measurability*, as it allows returning
 990 any function w' , not necessarily from the original witness class. Thus, if auditing is hard for a class
 991 $\mathcal{W}_{\text{target}}$, and if our class $\mathcal{W}(\mathcal{U})$ is at least as expressive as $\mathcal{W}_{\text{target}}$, then proactive measurability for \mathcal{U}
 992 is also computationally hard.

993 Gopalan et al. [11] demonstrate hardness for two key notions.

1. First, for decision calibration, their witness class \mathcal{W}_{dec} involves partitioning Δ^{C-1} using hyper-
 planes, i.e.

$$\mathcal{W}_{\text{dec}} := \{x \rightarrow g' \mathbb{1}\{a^T x \geq b\} + g \mathbb{1}\{a^T x < b\} \mid g', g, a \in \mathbb{R}^C, b \in \mathbb{R}, \text{ s.t. } \|g'\|_2, \|g\|_2 \leq 1\}.$$

994 Auditing for \mathcal{W}_{dec} is shown to be computationally hard under standard assumptions (Gopalan et al.
 995 [11, Thm. 5.1]), i.e. cannot be performed in polynomial time for non-trivial α under standard
 996 computational complexity assumptions. Up to a scaling, this is a slightly more general class
 997 than \mathcal{U}_{lin} .

998 2. Second, Gopalan et al. [11] introduced projected smooth calibration, which is very similar to
 999 our notion of utility calibration for \mathcal{U}_{lin} , but replaced the hard interval indicator with Lipschitz
 1000 functions. Again, auditing for this notion was also proven computationally hard, in the sense that
 1001 no auditing algorithm can be polynomial in $1/\alpha$ [11, Thm. 8.1].

1002 B.4 Interactive Measurability

1003 In this section, we establish conditions under which the utility calibration error $\text{UC}(f, \mathcal{U})$ is interac-
 1004 tively measurable, completing Section 3.4. This involves bounding the sample complexity required
 1005 for the empirical estimates $\widehat{\text{UC}}(f, u; S)$ to uniformly converge to their true values $\text{UC}(f, u)$ over all
 1006 $u \in \mathcal{U}$. For a given utility class \mathcal{U} , we define the class of functions

$$\mathcal{G}_{\mathcal{U}} := \{(X, Y) \mapsto \langle Y - f(X), \vec{u}(X) \mathbb{1}\{v_u(X) \in I\}\rangle \mid u \in \mathcal{U}, I \in \mathbb{I}[-1, 1]\}. \quad (\text{B.2})$$

1007 **Definition B.2** (Rademacher Complexity [43]). *Let \mathcal{F} be a class of real-valued functions $h : \mathcal{Z} \rightarrow \mathbb{R}$.
 1008 Given n samples $S_Z = (Z_1, \dots, Z_n)$ where $Z_i \sim D_Z$, the empirical Rademacher complexity of \mathcal{F}
 1009 given S_Z is*

$$\hat{\mathfrak{R}}_n(\mathcal{F}|S_Z) = \mathbb{E}_{\sigma} \left[\sup_{h \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^n \sigma_i h(Z_i) \right],$$

1010 where σ_i are i.i.d. Rademacher random variables. The expected Rademacher complexity is $\mathfrak{R}_n(\mathcal{F}) =$
 1011 $\mathbb{E}_{S_Z} [\hat{\mathfrak{R}}_n(\mathcal{F}|S_Z)]$.

1012 We establish the following theorem.

1013 **Theorem B.3** (Rademacher Complexity Bound for $\mathcal{G}_{\mathcal{U}}$). *Let \mathcal{U} be a class of utility functions. For
 1014 each $j \in [C]$, define the scalar-valued function class:*

$$\mathcal{P}_j(\mathcal{U}) := \{X \mapsto u(f(X), e_j) \mathbb{1}\{v_u(X) \in I\} \mid u \in \mathcal{U}, I \in \mathbb{I}[-1, 1]\}.$$

1015 Then, the Rademacher complexity of $\mathcal{G}_{\mathcal{U}}$ is bounded as:

$$\mathfrak{R}_n(\mathcal{G}_{\mathcal{U}}) \leq 2 \sum_{j=1}^C \mathfrak{R}_n(\mathcal{P}_j(\mathcal{U})).$$

1016 *Proof.* The class $\mathcal{G}_{\mathcal{U}}$ consists of functions $g_{u,I}(X, Y) = \langle Y - f(X), \vec{u}(X) \mathbb{1}\{v_u(X) \in I\}\rangle$. We
 1017 consider the empirical Rademacher complexity $\hat{\mathfrak{R}}_n(\mathcal{G}_{\mathcal{U}}|S_{XY})$ for n samples $S_{XY} = \{(X_k, Y_k)\}_{k=1}^n$,
 1018 which is by definition:

$$\hat{\mathfrak{R}}_n(\mathcal{G}_{\mathcal{U}}|S_{XY}) = \mathbb{E}_{\sigma'} \left[\sup_{\substack{u \in \mathcal{U} \\ I \in \mathbb{I}[-1, 1]}} \frac{1}{n} \sum_{k=1}^n \sigma'_k \langle Y_k - f(X_k), \vec{u}(f(X_k)) \mathbb{1}\{v_u(X_k) \in I\}\rangle \right],$$

1019 where σ'_k are i.i.d. scalar Rademacher random variables. For each $k \in [n]$, let $W_k = Y_k - f(X_k)$.
 1020 The function $\phi_{W_k} : \mathbb{R}^C \rightarrow \mathbb{R}$ defined by $\phi_{W_k}(z) = \langle W_k, z \rangle$ is L -Lipschitz and $\phi_{W_k}(\mathbf{0}) = 0$.
 1021 Specifically, for the ℓ_2 -norm on \mathbb{R}^C , the Lipschitz constant $L_k = \|W_k\|_2 = \|Y_k - f(X_k)\|_2 \leq \sqrt{2}$.
 1022 We take $L = \sqrt{2}$ as an upper bound for all k . Let $\vec{\mathcal{W}}_{\mathcal{U}}$ be the class of vector-valued functions from \mathcal{X}
 1023 to \mathbb{R}^C :

$$\vec{\mathcal{W}}_{\mathcal{U}} := \{X \mapsto \vec{u}(X) \mathbb{1}\{v_u(X) \in I\} \mid u \in \mathcal{U}, I \in \mathbb{I}[-1, 1]\}. \quad (\text{B.3})$$

1024 The functions $w \in \vec{\mathcal{W}}_{\mathcal{U}}$ map to $[-1, 1]^C$. Using Maurer [44, Corollary 1], we have

$$\hat{\mathfrak{R}}_n(\mathcal{G}_{\mathcal{U}}|S_{XY}) \leq \sqrt{2}L \cdot \hat{\mathfrak{R}}_n^{\text{vec}}(\vec{\mathcal{W}}_{\mathcal{U}}|S_X),$$

1025 where $\hat{\mathfrak{R}}_n^{\text{vec}}(\vec{\mathcal{W}}_{\mathcal{U}}|S_X)$ is the empirical vector Rademacher complexity of $\vec{\mathcal{W}}_{\mathcal{U}}$ given samples $S_X =$
 1026 $\{X_k\}_{k=1}^n$. It is defined as:

$$\hat{\mathfrak{R}}_n^{\text{vec}}(\vec{\mathcal{W}}_{\mathcal{U}}|S_X) := \mathbb{E}_{\sigma} \left[\sup_{w \in \vec{\mathcal{W}}_{\mathcal{U}}} \frac{1}{n} \sum_{k=1}^n \langle \sigma_k, w(X_k) \rangle \right], \quad (\text{B.4})$$

1027 where $\sigma_k \in \{-1, 1\}^C$ are vectors whose components σ_{kj} are i.i.d. Rademacher random variables.
 1028 Substituting $L = \sqrt{2}$:

$$\hat{\mathfrak{R}}_n(\mathcal{G}_U|S_{XY}) \leq \sqrt{2} \cdot \sqrt{2} \cdot \hat{\mathfrak{R}}_n^{\text{vec}}(\vec{\mathcal{W}}_U|S_X) = 2\hat{\mathfrak{R}}_n^{\text{vec}}(\vec{\mathcal{W}}_U|S_X).$$

1029 The vector Rademacher complexity $\hat{\mathfrak{R}}_n^{\text{vec}}(\vec{\mathcal{W}}_U|S_X)$ can be then bounded as:

$$\begin{aligned} \hat{\mathfrak{R}}_n^{\text{vec}}(\vec{\mathcal{W}}_U|S_X) &= \mathbb{E}_{\sigma} \left[\sup_{\substack{u \in \mathcal{U} \\ I \in \mathbb{I}[-1,1]}} \frac{1}{n} \sum_{k=1}^n \sum_{j=1}^C \sigma_{kj} u(f(X_k), e_j) \mathbb{1}\{v_u(X_k) \in I\} \right] \quad \text{by eq. (B.3) and (B.4)} \\ &\leq \mathbb{E}_{\sigma} \left[\sum_{j=1}^C \sup_{\substack{u \in \mathcal{U} \\ I \in \mathbb{I}[-1,1]}} \frac{1}{n} \sum_{k=1}^n \sigma_{kj} u(f(X_k), e_j) \mathbb{1}\{v_u(X_k) \in I\} \right] \quad (\text{subadditivity of sup}) \\ &= \sum_{j=1}^C \mathbb{E}_{\sigma \cdot j} \left[\sup_{\substack{u \in \mathcal{U} \\ I \in \mathbb{I}[-1,1]}} \frac{1}{n} \sum_{k=1}^n \sigma_{kj} u(f(X_k), e_j) \mathbb{1}\{v_u(X_k) \in I\} \right] \\ &= \sum_{j=1}^C \hat{\mathfrak{R}}_n(\mathcal{P}_j(\mathcal{U})|S_X). \end{aligned}$$

1030 Here $\sigma \cdot j$ denotes the j -th column of the matrix $\sigma = ((\sigma_{kj})_{k \in [n], j \in [C]})$. Combining these, we
 1031 get $\hat{\mathfrak{R}}_n(\mathcal{G}_U|S_{XY}) \leq 2 \sum_{j=1}^C \hat{\mathfrak{R}}_n(\mathcal{P}_j(\mathcal{U})|S_X)$. Taking expectation over S_{XY} yields the theorem
 1032 statement. \square

1033 **Corollary B.4** (Interactive Measurability from Rademacher Bound). *Let \mathcal{U} be a class of utility*
 1034 *functions and \mathcal{G}_U be the function class defined in eq. (B.2). With probability at least $1 - \delta$ over the*
 1035 *draw of $S \sim D^n$:*

$$\sup_{u \in \mathcal{U}} |\widehat{\text{UC}}(f, u; S) - \text{UC}(f, u)| \leq 2\mathfrak{R}_n(\mathcal{G}_U) + 4\sqrt{\frac{\log(2/\delta)}{2n}}.$$

1036 Combined with Theorem B.3, this implies:

$$\sup_{u \in \mathcal{U}} |\widehat{\text{UC}}(f, u; S) - \text{UC}(f, u)| \leq 4 \sum_{j=1}^C \mathfrak{R}_n(\mathcal{P}_j(\mathcal{U})) + 4\sqrt{\frac{\log(2/\delta)}{2n}}.$$

1037 *Proof.* For $u \in \mathcal{U}$ and $I \in \mathbb{I}[-1, 1]$, define $g_{u,I}(X, Y) := \langle Y - f(X), \vec{u}(X) \mathbb{1}\{v_u(X) \in I\} \rangle$. With
 1038 $\widehat{\text{UC}}(f, u; S)$ defined in Lemma 3.3, it holds by definition that

$$\begin{aligned} \sup_{u \in \mathcal{U}} |\widehat{\text{UC}}(f, u; S) - \text{UC}(f, u)| &= \sup_{u \in \mathcal{U}} \left| \sup_{I \in \mathbb{I}[-1,1]} |\hat{\mathbb{E}}_n[g_{u,I}(X, Y)]| - \sup_{I \in \mathbb{I}[-1,1]} |\mathbb{E}[g_{u,I}(X, Y)]| \right| \\ &\leq \sup_{u \in \mathcal{U}} \sup_{I \in \mathbb{I}[-1,1]} \left| |\hat{\mathbb{E}}_n[g_{u,I}(X, Y)]| - |\mathbb{E}[g_{u,I}(X, Y)]| \right| \\ &\leq \sup_{u \in \mathcal{U}} \sup_{I \in \mathbb{I}[-1,1]} \left| \hat{\mathbb{E}}_n[g_{u,I}(X, Y)] - \mathbb{E}[g_{u,I}(X, Y)] \right| \\ &= \sup_{g \in \mathcal{G}_U} \left| \hat{\mathbb{E}}_n[g(X, Y)] - \mathbb{E}[g(X, Y)] \right|. \end{aligned}$$

1039 The first inequality is by $\sup_x f(x) - \sup_x g(x) \leq \sup_x |(f - g)(x)|$. Then using [43, Theorem 3.3]
 1040 (standard symmetrization argument using Rademacher complexity and application of McDiarmid
 1041 inequality), with probability at least $1 - \delta$, it holds that

$$\sup_{g \in \mathcal{G}_U} \left| \hat{\mathbb{E}}_n[g(X, Y)] - \mathbb{E}[g(X, Y)] \right| \leq 2\mathfrak{R}_n(\mathcal{G}_U) + 4\sqrt{\frac{\log(2/\delta)}{2n}}.$$

1042 The corollary follows by substituting the bound for $\mathfrak{R}_n(\mathcal{G}_U)$ from Theorem B.3 into the equation
 1043 above. \square

1044 We now apply the statements established above to the case of linear utilities (Example 3.6).

1045 **Corollary B.5** (Interactive Measurability of \mathcal{U}_{lin}). *The utility calibration error is interactively*
 1046 *measurable for the class of linear utilities \mathcal{U}_{lin} (Example 3.6). The sample complexity is $N =$*
 1047 *$O\left(\frac{C^3 \log(n/C) + \log(1/\delta)}{\varepsilon^2}\right)$.*

1048 *Proof.* For $u_a \in \mathcal{U}_{\text{lin}}$, we have, by definition, $u_a(f(X), e_j) = a_j$ (see Example 3.6), where $a \in$
 1049 $[-1, 1]^C$. The predicted utility is $v_{u_a}(X) = \langle f(X), a \rangle$. The component classes $\mathcal{P}_j(\mathcal{U}_{\text{lin}})$ are:

$$\mathcal{P}_j(\mathcal{U}_{\text{lin}}) = \{X \mapsto a_j \mathbb{1}\{\langle f(X), a \rangle \in I\} \mid a \in [-1, 1]^C, I \in \mathbb{I}[-1, 1]\}.$$

1050 Each function in $\mathcal{P}_j(\mathcal{U}_{\text{lin}})$ is a product of a_j and an indicator function $\mathbb{1}\{\langle f(X), a \rangle \in I\}$. First, we
 1051 consider the class of functions $\mathcal{H}_{a,I} = \{X \mapsto \mathbb{1}\{\langle f(X), a \rangle \in I\} \mid a \in [-1, 1]^C\}$. $\mathcal{H}_{a,I}$ is a subclass
 1052 of the indicator functions of polytopes with two supporting hyperplanes. Thus, $\text{VC}(\mathcal{H}_{a,I}) = O(C)$

1053 [45, Theorem 1] and $\mathfrak{R}_n(\mathcal{H}_{a,I}) \leq O\left(\sqrt{\frac{C \log(n/C)}{n}}\right)$. We now proceed to bound $\mathfrak{R}_n(\mathcal{P}_j(\mathcal{U}_{\text{lin}}))$.

1054 Let $S_X = \{X_1, \dots, X_n\}$ be n i.i.d. samples. The empirical Rademacher complexity of $\mathcal{P}_j(\mathcal{U}_{\text{lin}})$ is:

$$\begin{aligned} \hat{\mathfrak{R}}_n(\mathcal{P}_j(\mathcal{U}_{\text{lin}}) | S_X) &= \mathbb{E}_{\sigma} \left[\sup_{\substack{a \in [-1, 1]^C \\ I \in \mathbb{I}[-1, 1]}} \frac{1}{n} \sum_{k=1}^n \sigma_k a_j \mathbb{1}\{\langle f(X_k), a \rangle \in I\} \right] \\ &\leq \mathbb{E}_{\sigma} \left[\sup_{\substack{a \in [-1, 1]^C \\ I \in \mathbb{I}[-1, 1]}} |a_j| \left| \frac{1}{n} \sum_{k=1}^n \sigma_k \mathbb{1}\{\langle f(X_k), a \rangle \in I\} \right| \right] \\ &\leq \mathbb{E}_{\sigma} \left[\sup_{\substack{a \in [-1, 1]^C \\ I \in \mathbb{I}[-1, 1]}} \left| \frac{1}{n} \sum_{k=1}^n \sigma_k \mathbb{1}\{\langle f(X_k), a \rangle \in I\} \right| \right] \quad (\text{since } |a_j| \leq 1) \\ &= \mathbb{E}_{\sigma} \left[\sup_{\substack{h \in \mathcal{H}_{a,I} \\ \xi \in \{-1, 1\}}} \frac{1}{n} \sum_{k=1}^n \xi \sigma_k h(X_k) \right] \\ &\leq \mathbb{E}_{\sigma} \left[\sup_{h \in \mathcal{H}_{a,I}} \frac{1}{n} \sum_{k=1}^n \sigma_k h(X_k) \right] + \mathbb{E}_{\sigma} \left[\sup_{h \in \mathcal{H}_{a,I}} \frac{1}{n} \sum_{k=1}^n -\sigma_k h(X_k) \right] \\ &\leq 2\hat{\mathfrak{R}}_n(\mathcal{H}_{a,I} | S_X). \end{aligned}$$

1055 Thus, taking expectations over S_X :

$$\mathfrak{R}_n(\mathcal{P}_j(\mathcal{U}_{\text{lin}})) \leq 2\mathfrak{R}_n(\mathcal{H}_{a,I}) = O\left(\sqrt{\frac{C \log(n/C)}{n}}\right).$$

1056 Using Corollary B.4, the uniform error bound is:

$$\begin{aligned} \sup_{u \in \mathcal{U}_{\text{lin}}} |\widehat{\text{UC}}(f, u; S) - \text{UC}(f, u)| &\leq 4 \sum_{j=1}^C \mathfrak{R}_n(\mathcal{P}_j(\mathcal{U}_{\text{lin}})) + 4\sqrt{\frac{\log(2/\delta)}{2n}} \\ &= O\left(\sqrt{\frac{C^3 \log(n/C)}{n}} + \frac{\log(1/\delta)}{n}\right). \end{aligned}$$

1057 □

1058 **Corollary B.6** (Interactive Measurability of $\mathcal{U}_{\text{rank}}$). *The utility calibration error is interactively*
 1059 *measurable for the class of rank-based utilities $\mathcal{U}_{\text{rank}}$ (Example 3.7). The sample complexity is*
 1060 *$N = O\left(\frac{C^3 \log(n/C) + \log(1/\delta)}{\varepsilon^2}\right)$.*

1061 *Proof.* First, we demonstrate that functions in $\mathcal{G}_{\mathcal{U}_{\text{rank}}}$ can be expressed in a form analogous to those
 1062 in $\mathcal{G}_{\mathcal{U}_{\text{lin}}}$ by applying a rank-based transformation to the inputs $f(X)$ and Y .
 1063 Let $p = f(X)$. Define $\pi_p : [C] \rightarrow [C]$ as the permutation sorting p 's components in descending order,
 1064 so $p_{\pi_p(s)}$ is the s -th largest confidence. This means $\text{rank}(p, \pi_p(s)) = s$. Define the rank-transformed
 1065 prediction $\tilde{f}(X) \in \mathbb{R}^C$ by $\tilde{f}(X)_s = f(X)_{\pi_p(X)(s)}$, and the rank-transformed label $\tilde{Y}(X) \in \mathbb{R}^C$ by
 1066 $\tilde{Y}(X)_s = Y_{\pi_p(X)(s)}$. A utility function $u_\theta \in \mathcal{U}_{\text{rank}}$ is $u_\theta(p, e_j) = \theta_{\text{rank}(p, j)}$. The predicted utility
 1067 is $v_{u_\theta}(X) = \langle \tilde{f}(X), \theta \rangle$. The witness vector component is $\vec{u}_\theta(X)_j = \theta_{\text{rank}(f(X), j)}$. A function
 1068 $g \in \mathcal{G}_{\mathcal{U}_{\text{rank}}}$ is $(X, Y) \mapsto \langle Y - f(X), \vec{u}_\theta(X) \mathbb{1}\{v_{u_\theta}(X) \in I\} \rangle$. This can be rewritten as:

$$\begin{aligned} \langle Y - f(X), \vec{u}_\theta(X) \mathbb{1}\{v_{u_\theta}(X) \in I\} \rangle &= \left(\sum_{s=1}^C (Y_{\pi_p(X)(s)} - f(X)_{\pi_p(X)(s)}) \theta_s \right) \mathbb{1}\left\{ \langle \tilde{f}(X), \theta \rangle \in I \right\} \\ &= \langle \tilde{Y}(X) - \tilde{f}(X), \theta \cdot \mathbb{1}\left\{ \langle \tilde{f}(X), \theta \rangle \in I \right\} \rangle. \end{aligned}$$

This is equivalent to $\mathcal{G}_{\mathcal{U}_{\text{lin}}}$ up to a fixed data preprocessing step. By repeating the same steps as Corollary B.5, we recover

$$\mathfrak{R}_n(\mathcal{G}_{\mathcal{U}_{\text{rank}}}) = O\left(\sqrt{\frac{C^3 \log(n/C)}{n}}\right),$$

1069 which implies the stated sample complexity $N = O\left(\frac{C^3 \log(N/C) + \log(1/\delta)}{\epsilon^2}\right)$.

1070 □

1071 B.5 Quantile Guarantees

1072 The evaluation methodology in Section 4 relies on plotting the empirical CDF $\hat{F}_{E, M, n}$, constructed
 1073 from M utility functions and n data points. $\hat{F}_{E, M, n}$ serves as a proxy for the true CDF F_E of utility
 1074 calibration errors. Theorem B.7 bounds the distance between the curves, as characterized by the
 1075 L_2 -distance. While weaker than characterizing the deviation using L_∞ -norm, L_2 distance allows a
 1076 bound without any assumptions on the smoothness of the underlying CDF F_E . The bound depends on
 1077 two factors: first, the uniform accuracy ϵ_{stat} with which individual utility calibration errors $\text{UC}(f, u)$
 1078 can be estimated by $\widehat{\text{UC}}(f, u; S)$, and second, the number of utility functions M sampled to construct
 1079 the $\hat{F}_{E, M, n}$.

1080 **Theorem B.7.** Let $F_E(e) := \mathbb{P}_{u \sim \mathcal{D}_{\mathcal{U}}}(\text{UC}(f, u) \leq e)$ be the true CDF of utility calibration
 1081 errors, where u is drawn from a distribution $\mathcal{D}_{\mathcal{U}}$ over the utility class \mathcal{U} . Let $\hat{F}_{E, M, n}(e) :=$
 1082 $\frac{1}{M} \sum_{m=1}^M \mathbb{1}\{\widehat{\text{UC}}(f, u_m; S) \leq e\}$ be its empirical estimate, based on M i.i.d. utility functions
 1083 $u_1, \dots, u_M \sim \mathcal{D}_{\mathcal{U}}$ and a data sample S of size n ($\delta_S, \epsilon_{\text{stat}}$). Assume that, with probability at least
 1084 $1 - \delta_S$ over the draw of S ,

$$\sup_{u \in \mathcal{U}} |\widehat{\text{UC}}(f, u; S) - \text{UC}(f, u)| \leq \epsilon_{\text{stat}}.$$

1085 Then, with probability at least $(1 - \delta_S - \delta_M)$ over the draw of S and $\{u_m\}_{m=1}^M$, it holds that

$$\|F_E - \hat{F}_{E, M, n}\|_{L^2([0, 2])} \leq \sqrt{2\epsilon_{\text{stat}}} + \sqrt{\frac{\ln(2/\delta_M)}{M}}.$$

1086 *Proof.* Let $E(u) := \text{UC}(f, u)$ be the true utility calibration error for a utility function $u \sim \mathcal{D}_{\mathcal{U}}$, and
 1087 let $\hat{E}_S(u) := \widehat{\text{UC}}(f, u; S)$ be its empirical estimate based on data sample S . The CDF of $E(u)$ is
 1088 F_E , and let $F_{\hat{E}_S}$ be the CDF of $\hat{E}_S(u)$ when $u \sim \mathcal{D}_{\mathcal{U}}$ and S is fixed.

1089 We condition on the event (occurring with probability at least $1 - \delta_S$) that S is such that $|E(u) -$
 1090 $\hat{E}_S(u)| \leq \epsilon_{\text{stat}}$ for all $u \in \mathcal{U}$. This implies that for any u , $E(u) - \epsilon_{\text{stat}} \leq \hat{E}_S(u) \leq E(u) + \epsilon_{\text{stat}}$.

1091 Consequently, for any $u \in \mathcal{U}$, as $E(u)$ and $\hat{E}(u)$ belong to the set $[0, 2]$, it holds for any $e \in [0, 2]$
 1092 that

$$\begin{aligned} F_E(e - \varepsilon_{\text{stat}}) &= \mathbb{P}(E(u) \leq e - \varepsilon_{\text{stat}}) \leq \mathbb{P}(\hat{E}_S(u) \leq e) = F_{\hat{E}_S}(e) \\ F_{\hat{E}_S}(e) &= \mathbb{P}(\hat{E}_S(u) \leq e) \leq \mathbb{P}(E(u) \leq e + \varepsilon_{\text{stat}}) = F_E(e + \varepsilon_{\text{stat}}). \end{aligned}$$

1093 Thus, $F_E(e - \varepsilon_{\text{stat}}) \leq F_{\hat{E}_S}(e) \leq F_E(e + \varepsilon_{\text{stat}})$. This implies that

$$|F_{\hat{E}_S}(e) - F_E(e)| \leq \max\{F_E(e + \varepsilon_{\text{stat}}) - F_E(e), F_E(e) - F_E(e - \varepsilon_{\text{stat}})\}.$$

1094 Let $\Delta_E(e)$ denote the right-hand side. The L^2 distance squared between $F_{\hat{E}_S}$ and F_E is

$$\begin{aligned} \|F_{\hat{E}_S} - F_E\|_{L^2([0,2])}^2 &= \int_0^2 (F_{\hat{E}_S}(e) - F_E(e))^2 de \\ &\leq \int_0^2 \Delta_E(e)^2 de \leq \int_0^2 \Delta_E(e) de, \end{aligned}$$

1095 since $0 \leq \Delta_E(e) \leq 1$. Further, as $\max(a, b) \leq a + b$ for non-negative a, b , it follows that

$$\int_0^2 \Delta_E(e) de \leq \int_0^2 [F_E(e + \varepsilon_{\text{stat}}) - F_E(e)] de + \int_0^2 [F_E(e) - F_E(e - \varepsilon_{\text{stat}})] de.$$

1096 The first term evaluates to $\int_2^{2+\varepsilon_{\text{stat}}} F_E(v) dv - \int_0^{\varepsilon_{\text{stat}}} F_E(v) dv$. Since $F_E(v) = 1$ for $v \geq 2$ and
 1097 $F_E(v) \geq 0$, this term is $\leq \varepsilon_{\text{stat}}$. The second term evaluates to $\int_{2-\varepsilon_{\text{stat}}}^2 F_E(v) dv - \int_{-\varepsilon_{\text{stat}}}^0 F_E(v) dv$.
 1098 Since $F_E(v) = 0$ for $v < 0$ (errors are non-negative) and $F_E(v) \leq 1$, this term is $\leq \varepsilon_{\text{stat}}$. Thus,
 1099 $\int_0^2 \Delta_E(e) de \leq 2\varepsilon_{\text{stat}}$, which implies $\|F_{\hat{E}_S} - F_E\|_{L^2([0,2])} \leq \sqrt{2\varepsilon_{\text{stat}}}$.

1100 Next, $\hat{F}_{E,M,n}(e)$ is the empirical CDF of M i.i.d. samples $\{\hat{E}_S(u_m)\}_{m=1}^M$ drawn according to $F_{\hat{E}_S}$.
 1101 By the Dvoretzky–Kiefer–Wolfowitz (DKW) inequality [46], with probability at least $1 - \delta_M$:

$$\|\hat{F}_{E,M,n} - F_{\hat{E}_S}\|_{\infty} \leq \sqrt{\frac{\ln(2/\delta_M)}{2M}}.$$

1102 The L^2 distance can be bounded by the L^{∞} distance, as

$$\begin{aligned} \|F_{\hat{E}_S} - \hat{F}_{E,M,n}\|_{L^2([0,2])}^2 &= \int_0^2 (F_{\hat{E}_S}(e) - \hat{F}_{E,M,n}(e))^2 de \\ &\leq \int_0^2 \|\hat{F}_{E,M,n} - F_{\hat{E}_S}\|_{\infty}^2 de \\ &= 2\|\hat{F}_{E,M,n} - F_{\hat{E}_S}\|_{\infty}^2 \leq \frac{\ln(2/\delta_M)}{M}. \end{aligned}$$

1103 So, $\|F_{\hat{E}_S} - \hat{F}_{E,M,n}\|_{L^2([0,2])} \leq \sqrt{\frac{\ln(2/\delta_M)}{M}}$. Finally, by the triangle inequality, combining the two
 1104 bounds:

$$\begin{aligned} \|F_E - \hat{F}_{E,M,n}\|_{L^2([0,2])} &\leq \|F_E - F_{\hat{E}_S}\|_{L^2([0,2])} + \|F_{\hat{E}_S} - \hat{F}_{E,M,n}\|_{L^2([0,2])} \\ &\leq \sqrt{2\varepsilon_{\text{stat}}} + \sqrt{\frac{\ln(2/\delta_M)}{M}}. \end{aligned}$$

1105 Using a union bound, this holds with probability at least $(1 - \delta_S - \delta_M)$. \square

1106 C Proofs

1107 Before proceeding to the proofs of the statements in the main paper, we list a common result
 1108 combining Massart’s lemma, which bounds the Rademacher complexity using the growth function
 1109 [43, Theorem 3.7] and Sauer’s lemma, which bounds the growth function [43, Theorem 3.17].

1110 **Result 1.** Consider a boolean class of functions \mathcal{B} such that $d = VC(\mathcal{B})$. Let $d > n$, then it holds
 1111 that

$$\mathfrak{R}_n(\mathcal{B}) \leq \sqrt{\frac{2d \log(en/d)}{n}}.$$

1112 C.1 Proof of Proposition 3.1

1113 *Proof.* At a high-level, the proof is similar to the approach of Rossellini et al. [23] but some details
1114 differ. We start by analyzing the loss function

$$\ell_{\text{util}}(u_Y, \hat{U}; t_0) = (t_0 - u_Y)(\hat{U} - \mathbb{1}\{u_Y \geq t_0\}). \quad (\text{C.1})$$

- 1115 • If $\hat{U} = \mathbb{1}\{u_Y \geq t_0\}$, then $\ell_{\text{util}} = 0$.
- 1116 • If $\hat{U} = 1$ and $u_Y < t_0$, the loss is $t_0 - u_Y > 0$.
- 1117 • If $\hat{U} = 0$ but $u_Y \geq t_0$, the loss is $u_Y - t_0 \geq 0$.

1118 The loss, which is only non-zero when a mismatch $\hat{U} \neq \mathbb{1}\{u + Y \geq t_0\}$ occurs, is $|u_Y - t_0|$. This loss
1119 function appropriately penalizes mismatches between the action taken based on $v_u(X)$ and the ideal
1120 action based on u_Y . We now consider any monotone non-decreasing function $h : [-1, 1] \rightarrow [-1, 1]$.
1121 The difference in risks between using $v_u(X)$ directly and using $h(v_u(X))$ is

$$\begin{aligned} \Delta R &= R_{\text{util}}(v_u(X); t_0) - R_{\text{util}}(h(v_u(X)); t_0) \\ &= \mathbb{E}\left[(t_0 - u_Y)(\mathbb{1}\{v_u(X) \geq t_0\} - \mathbb{1}\{u_Y \geq t_0\})\right] - \mathbb{E}\left[(t_0 - u_Y)(\mathbb{1}\{h(v_u(X)) \geq t_0\} - \mathbb{1}\{u_Y \geq t_0\})\right] \\ &= \mathbb{E}\left[(t_0 - u_Y)(\mathbb{1}\{v_u(X) \geq t_0\} - \mathbb{1}\{h(v_u(X)) \geq t_0\})\right]. \end{aligned}$$

1122 Let $E_1 = \{X \mid v_u(X) < t_0, h(v_u(X)) \geq t_0\}$ and $E_2 = \{X \mid v_u(X) \geq t_0, h(v_u(X)) < t_0\}$. The
1123 term $\mathbb{1}\{v_u(X) \geq t_0\} - \mathbb{1}\{h(v_u(X)) \geq t_0\}$ equals -1 on E_1 and 1 on E_2 , and 0 elsewhere.

$$\begin{aligned} \Delta R &= \mathbb{E}\left[(t_0 - u_Y)(-\mathbb{1}\{X \in E_1\}) + (t_0 - u_Y)(\mathbb{1}\{X \in E_2\})\right] \\ &= \mathbb{E}\left[(u_Y - t_0)\mathbb{1}\{X \in E_1\}\right] - \mathbb{E}\left[(u_Y - t_0)\mathbb{1}\{X \in E_2\}\right] \\ &= \mathbb{E}\left[(u_Y - v_u(X) + v_u(X) - t_0)\mathbb{1}\{X \in E_1\}\right] - \mathbb{E}\left[(u_Y - v_u(X) + v_u(X) - t_0)\mathbb{1}\{X \in E_2\}\right] \\ &= \underbrace{\mathbb{E}\left[(u_Y - v_u(X))\mathbb{1}\{X \in E_1\}\right]}_A + \underbrace{\mathbb{E}\left[(v_u(X) - t_0)\mathbb{1}\{X \in E_1\}\right]}_C \\ &\quad - \underbrace{\mathbb{E}\left[(u_Y - v_u(X))\mathbb{1}\{X \in E_2\}\right]}_B - \underbrace{\mathbb{E}\left[(v_u(X) - t_0)\mathbb{1}\{X \in E_2\}\right]}_D. \end{aligned}$$

On E_1 , we have $v_u(X) < t_0$, which implies $v_u(X) - t_0 < 0$, so $C \leq 0$. On E_2 , we have $v_u(X) \geq t_0$, which implies $v_u(X) - t_0 \geq 0$, so $D \geq 0$. Thus, $\Delta R = A + C - B - D \leq A - B$.

$$A - B = \mathbb{E}[(u_Y - v_u(X))\mathbb{1}\{X \in E_1\}] + \mathbb{E}[(v_u(X) - u_Y)\mathbb{1}\{X \in E_2\}].$$

Since h is monotone non-decreasing, the sets E_1 and E_2 correspond to $v_u(X)$ lying within specific intervals (or unions of intervals which can be decomposed). Let I_1 be the set of $v_u(X)$ values defining E_1 (e.g., $v_u(X) < t_0$ and $h(v_u(X)) \geq t_0$) and I_2 be the set for E_2 . The terms $\mathbb{1}\{X \in E_1\}$ and $\mathbb{1}\{X \in E_2\}$ effectively restrict the expectation to regions where $v_u(X)$ falls into certain ranges. By the definition of $\text{UC}(f, u)$, for $E \in \{E_1, E_2\}$

$$\mathbb{E}[(u_Y - v_u(X))\mathbb{1}\{X \in E\}] \leq \sup_{I \in \mathbb{I}[-1, 1]} |\mathbb{E}[(u_Y - v_u(X))\mathbb{1}\{v_u(X) \in I\}]| \leq \text{UC}(f, u).$$

1124 Therefore, $\Delta R \leq A - B \leq \text{UC}(f, u) + \text{UC}(f, u) = 2\text{UC}(f, u)$. Since this holds for any monotone
1125 non-decreasing function h , taking the supremum over monotone functions completes the proof. \square

1126 C.2 Proof of Proposition 3.2

1127 *Proof.* The proof is the same as [23, Lemma A.2.], we include it for completeness. Assume
1128 $\text{UC}(f, u) > 0$. Let $U_Y := u(f(X), Y)$ denote the realized utility. Both U_Y and $v_u(X)$ take
1129 values in $[-1, 1]$.

1130 Let $W \in (0, 2]$ be a chosen bin width. We partition the interval $[-1, 1]$ into $K_W = \lceil 2/W \rceil$ disjoint
1131 intervals A_1, A_2, \dots, A_{K_W} . These intervals are constructed as follows: For $j = 1, \dots, K_W - 1$,

1132 let $A_j = [-1 + (j-1)W, -1 + jW]$. For $j = K_W$, let $A_{K_W} = [-1 + (K_W - 1)W, 1]$. This
 1133 construction ensures that $\bigcup_{j=1}^{K_W} A_j = [-1, 1]$, and each interval A_j has length $\lambda(A_j) \leq W$.

1134 Let $\psi_W : [-1, 1] \rightarrow \{1, \dots, K_W\}$ be the function mapping a value $z \in [-1, 1]$ to the index
 1135 j of the bin A_j such that $z \in A_j$. We construct a candidate calibrated predictor $g_W(X) :=$
 1136 $\mathbb{E}[U_Y \mid \psi_W(v_u(X))]$. The function $g_W(X)$ is perfectly calibrated: $\mathbb{E}[U_Y \mid g_W(X)] = \mathbb{E}[\mathbb{E}[U_Y \mid$
 1137 $\psi_W(v_u(X))] \mid g_W(X)]$. Since $g_W(X)$ is, by definition, measurable with respect to the sigma-
 1138 algebra generated by $\psi_W(v_u(X))$, it follows from the properties of conditional expectation that
 1139 $\mathbb{E}[U_Y \mid g_W(X)] = g_W(X)$ almost surely.

By the definition of DCU(f, u), which is the infimum of distances $\mathbb{E}|g(X) - v_u(X)|$ over all perfectly calibrated predictors $g(X)$, it holds that

$$\text{DCU}(f, u) \leq \mathbb{E}|g_W(X) - v_u(X)|.$$

To analyze the right-hand side, we introduce an intermediate term $V_{\text{avgbin}}(X) := \mathbb{E}[v_u(X) \mid \psi_W(v_u(X))]$. This represents the average of $v_u(X)$ within the bin $A_{\psi_W(v_u(X))}$ where $v_u(X)$ falls. Using the triangle inequality:

$$\mathbb{E}|g_W(X) - v_u(X)| \leq \mathbb{E}|g_W(X) - V_{\text{avgbin}}(X)| + \mathbb{E}|V_{\text{avgbin}}(X) - v_u(X)|.$$

1140 To bound the second term, $\mathbb{E}|V_{\text{avgbin}}(X) - v_u(X)|$: For any realization $X = x$, $v_u(x)$ is a point
 1141 in some bin A_j . $V_{\text{avgbin}}(x)$ is the conditional expectation of $v_u(X')$ given that $v_u(X')$ is in A_j .
 1142 As such, $V_{\text{avgbin}}(x)$ must also lie within the convex hull of A_j . Thus, $|V_{\text{avgbin}}(x) - v_u(x)| \leq W$.
 1143 Taking the expectation over X , we get $\mathbb{E}|V_{\text{avgbin}}(X) - v_u(X)| \leq W$.

1144 To bound the first term:

$$\begin{aligned} \mathbb{E}|g_W(X) - V_{\text{avgbin}}(X)| &= \mathbb{E}|\mathbb{E}[U_Y \mid \psi_W(v_u(X))] - \mathbb{E}[v_u(X) \mid \psi_W(v_u(X))]| \\ &= \mathbb{E}|\mathbb{E}[U_Y - v_u(X) \mid \psi_W(v_u(X))]| \\ &= \sum_{j=1}^{K_W} \mathbb{P}\{\psi_W(v_u(X)) = j\} |\mathbb{E}[U_Y - v_u(X) \mid \psi_W(v_u(X)) = j]| \\ &= \sum_{j=1}^{K_W} |\mathbb{E}[(U_Y - v_u(X)) \mathbb{1}\{\psi_W(v_u(X)) = j\}]| \\ &= \sum_{j=1}^{K_W} |\mathbb{E}[(U_Y - v_u(X)) \mathbb{1}\{v_u(X) \in A_j\}]|. \end{aligned}$$

By the definition of utility calibration error UC(f, u), for each bin A_j , we have:

$$|\mathbb{E}[(U_Y - v_u(X)) \mathbb{1}\{v_u(X) \in A_j\}]| \leq \sup_{I \in \mathcal{I}[-1, 1]} |\mathbb{E}[(U_Y - v_u(X)) \mathbb{1}\{v_u(X) \in I\}]| = \text{UC}(f, u).$$

Therefore,

$$\mathbb{E}|g_W(X) - V_{\text{avgbin}}(X)| \leq \sum_{j=1}^{K_W} \text{UC}(f, u) = K_W \cdot \text{UC}(f, u).$$

1145 Since $K_W = \lceil 2/W \rceil$, and for any $x > 0$, $\lceil x \rceil \leq x + 1$ (with strict inequality if x is not an integer),
 1146 we have $K_W \leq 2/W + 1$. Thus, $\mathbb{E}|g_W(X) - V_{\text{avgbin}}(X)| \leq (2/W + 1) \cdot \text{UC}(f, u)$.

Combining the bounds for the two terms, we get:

$$\text{DCU}(f, u) \leq \left(\frac{2}{W} + 1 \right) \cdot \text{UC}(f, u) + W.$$

1147 This inequality holds for any chosen bin width $W \in (0, 2]$. Our goal is to select W to minimize
 1148 this upper bound. Set $W = W_{\text{opt}} := \sqrt{2\text{UC}(f, u)}$, noting that W_{opt} is in the domain $(0, 2]$ as
 1149 $\text{UC}(f, u) \leq 2$, for any u and f .

1150 Substituting $W_{\text{opt}} = \sqrt{2\text{UC}(f, u)}$ into the upper bound for DCU(f, u):

$$\begin{aligned} \text{DCU}(f, u) &\leq \frac{2}{\sqrt{2\text{UC}(f, u)}} \text{UC}(f, u) + \text{UC}(f, u) + \sqrt{2\text{UC}(f, u)} \\ &= 2\sqrt{2\text{UC}(f, u)} + \text{UC}(f, u). \end{aligned}$$

1152 C.3 Proof of Lemma 3.3

1153 *Proof.* Let $A(X, Y) := u(f(X), Y) - v_u(X)$. Since $u(\cdot, \cdot) \in [-1, 1]$, $v_u(X) \in [-1, 1]$, so
 1154 $A(X, Y) \in [-2, 2]$. Define the function class

$$\mathcal{F}_{uc} = \{h_I : (X, Y) \mapsto A(X, Y) \cdot \mathbb{1}\{v_u(X) \in I\} \mid I \in \mathbb{I}[-1, 1]\}.$$

1155 The true and empirical utility calibration errors are $\text{UC}(f, u) = \sup_{h \in \mathcal{F}_{uc}} |\mathbb{E}[h(X, Y)]|$ and
 1156 $\widehat{\text{UC}}(f, u; S) = \sup_{h \in \mathcal{F}_{uc}} |\hat{\mathbb{E}}_n[h]|$, where $\hat{\mathbb{E}}_n[h] = \frac{1}{n} \sum_{i=1}^n h(X_i, Y_i)$. We bound $\sup_{h \in \mathcal{F}_{uc}} |\mathbb{E}[h] - \hat{\mathbb{E}}_n[h]|$.
 1157

1158 Let $\mathcal{G}_{int} = \{X \mapsto \mathbb{1}\{v_u(X) \in I\} \mid I \in \mathbb{I}[-1, 1]\}$. Since $v_u(X)$ is a fixed function and the class of
 1159 interval indicators on \mathbb{R} has VC dimension 2, $VC(\mathcal{G}_{int}) \leq 2$. By Result 1, for $d = VC(\mathcal{G}_{int}) = 2$
 1160 (assuming $n \geq 2$):

$$\mathfrak{R}_n(\mathcal{G}_{int}) \leq \sqrt{\frac{2 \cdot 2 \log(en/2)}{n}} = \sqrt{\frac{4 \log(en/2)}{n}}.$$

1161 The functions in \mathcal{F}_{uc} are $h_I(X, Y) = A(X, Y) \cdot g_I(X)$ where $g_I(X) \in \mathcal{G}_{int}$. For a fixed f and u ,
 1162 $A(X, Y)$ is a fixed function with $|A(X, Y)| \leq 2$. By [43, Lemma 5.7] with $l = 2$:

$$\mathfrak{R}_n(\mathcal{F}_{uc}) \leq 2 \cdot \mathfrak{R}_n(\mathcal{G}_{int}) \leq 2\sqrt{\frac{4 \log(en/2)}{n}} = 4\sqrt{\frac{\log(en/2)}{n}}.$$

1163 Functions $h \in \mathcal{F}_{uc}$ are bounded in $[-2, 2]$, so their range is $M = 4$. Using a standard Rademacher
 1164 symmetrization argument and McDiarmid's inequality (e.g., [47, Page 93]), for any $\delta > 0$, with
 1165 probability at least $1 - \delta$:

$$\begin{aligned} \sup_{h \in \mathcal{F}_{uc}} |\mathbb{E}[h] - \hat{\mathbb{E}}_n[h]| &\leq 2\mathfrak{R}_n(\mathcal{F}_{uc}) + M\sqrt{\frac{\log(1/\delta)}{2n}} \\ &\leq 2 \left(4\sqrt{\frac{\log(en/2)}{n}} \right) + 4\sqrt{\frac{\log(1/\delta)}{2n}} \\ &= 8\sqrt{\frac{\log(en/2)}{n}} + 2\sqrt{\frac{2 \log(1/\delta)}{n}}. \end{aligned}$$

1166 *Computational Complexity:* To compute $\widehat{\text{UC}}(f, u; S)$: For $i = 1, \dots, n$, compute $V_i = v_u(X_i)$ and
 1167 $A_i = u(f(X_i), Y_i) - V_i$. This is $O(n \cdot T_{eval})$. Sort pairs (V_i, A_i) by V_i to get $(V_{(i)}, A_{(i)})$: $O(n \log n)$.
 1168 Compute prefix sums $P_k = \sum_{j=1}^k A_{(j)}$: $O(n)$. The supremum $\sup_I |\frac{1}{n} \sum A_j \mathbb{1}\{V_j \in I\}|$ is found
 1169 by checking $O(n^2)$ sums $\sum_{k=j}^l A_{(k)} = P_l - P_{j-1}$ (for $1 \leq j \leq l \leq n$, plus intervals starting at -1
 1170 or ending at 1). This takes $O(n^2)$. Total time: $O(n \cdot T_{eval} + n^2)$. □

1171 D Additional Experiments

1172 **Experimental Setup:** Our experiments were performed using standard image classification bench-
 1173 marks: CIFAR10, CIFAR100 [38, MIT License], and ImageNet-1K [37, Provided for non-commercial
 1174 use]. We used publicly available pretrained models. For each model-dataset combination, the valida-
 1175 tion data was divided into a calibration set (70%), for training post-hoc methods, and a test set (30%),
 1176 for evaluation. We repeated the experiments across 5 different calibration/test splits and reported the
 1177 average results. For CIFAR10/100 datasets, we used the model checkpoints available in [48]. For
 1178 ImageNet-1K, we used `timm` checkpoints [49].

1179 **Calibration Methods and Evaluation:** We benchmarked several well-known post-hoc calibration
 1180 techniques. These include Temperature Scaling (T.S.) [26, 7], Vector Scaling (V.S.) [41, 7], Ensemble
 1181 Temperature Scaling (Ens. T.S.) [42], Dirichlet recalibration [27], and Isotonic Regression (I.R.) [28],
 1182 which was applied both globally and in a one-vs-all (OvA) manner. For these methods, we based our
 1183 implementation on the one publicly available in [50]. For Dirichlet recalibration, we used the version

with L_2 regularization, with the regularization being tuned over 15 runs using BayesSearchCV of scikit-optimize [51] with the goal of minimizing the Brier score.

In addition, we implemented the patching style post-hoc calibration algorithm (Algorithm 1), referred to as Patch. We used the utility class $\mathcal{U} = \mathcal{U}_{\text{CWE}} \cup \mathcal{U}_{\text{topK}}$ (Example 3.5 and Example 3.7). We found that the theoretical stepsize in Proposition A.1 to not work well in practice and the algorithm to be sensitive to stepsize choice. This observation was also recently reported in [52]. Instead, we tuned the stepsize η in the grid $[0.005, 0.01, 0.05, 0.1, 0.2]$. We ran the algorithm for 125 steps. All post-hoc methods were trained on the designated calibration set and subsequently evaluated on the test set. The experiments took approximately 80 hours of A100 80G NVIDIA GPU and 100 CPU hours.

Evaluation. Model performance and calibration were assessed using a suite of metrics. Standard evaluations included Accuracy and Brier Score. For binned binarized approaches, we compute TCE^{bin} (2.1) and CWE^{bin} (2.2), using 15 equal-weight bins (each bin has the same number of datapoints). Furthermore, we evaluated our proposed binning-free utility calibration metrics for specific utility classes: \mathcal{U}_{TCE} , \mathcal{U}_{CWE} , and $\mathcal{U}_{\text{topK}}$.

As discussed in Section 4, to evaluate calibration over a broader range of diverse utility functions, we also generated and plotted the empirical Cumulative Distribution Functions (eCDFs) across the classes \mathcal{U}_{lin} and $\mathcal{U}_{\text{rank}}$. For each sampled utility function u , we used a batch of 5000 datapoints to calculate $\text{UC}(f, u)$. We report the eCDF curves using 500 sampled utility functions.

The results are thus given in tables and figures as follows.

Metrics	Results given in		
	CIFAR10	CIFAR100	ImageNet
TCE^{bin} , CWE^{bin} , \mathcal{U}_{TCE} , \mathcal{U}_{CWE} , and $\mathcal{U}_{\text{topK}}$	Table 2	Table 3	Table 4
\mathcal{U}_{lin} and $\mathcal{U}_{\text{rank}}$	Figure 2	Figure 3	Figure 4

CIFAR10/100 Tables. We provide experimental results for the CIFAR10 and CIFAR100 using RepVGG [53] and ShuffleNet [54] models. With the exception of Dirichlet, which degraded the model accuracy, most post-hoc algorithms reduced the calibration error across different calibration metrics without an effect on accuracy. In addition, we note that, even for uncalibrated models, class-wise calibration metrics (\mathcal{U}_{CWE} and CWE^{bin}) were small. In some cases, post-hoc methods degraded the class-wise metrics. Nonetheless, the miscalibration of the original models was most pronounced for the top-class and confidence-based metrics, i.e. $\mathcal{U}_{\text{topK}}$, \mathcal{U}_{TCE} , and TCE^{bin} , and most post-hoc methods effectively decreased them. We note that the best performing post-hoc method significantly varied across different metrics. In particular, Patch was the most effective at decreasing $\mathcal{U}_{\text{topK}}$. On the other hand, we observed poor performance when using Dirichlet calibration.

ImageNet-1K Tables. For the large-scale ImageNet-1K dataset, we evaluated the EfficientNet [55] and ViT_Base_P16 [40] models. The results shared common trends with CIFAR10/100. In particular, uncalibrated versions of these models exhibited significant miscalibration across the metrics $\mathcal{U}_{\text{topK}}$, \mathcal{U}_{TCE} , and TCE^{bin} . In addition, the calibration error for the class-wise metrics (\mathcal{U}_{CWE} and CWE^{bin}) was generally small and was occasionally worsened by post-hoc methods. Meanwhile, post-hoc methods effectively reduced the metrics $\mathcal{U}_{\text{topK}}$, \mathcal{U}_{TCE} , and TCE^{bin} . We note a large disparity between the performance of Patch across the two models, indicating that the patching style algorithms can be sensitive to the setting. We also note that Patch was generally sensitive to its hyperparameters.

CIFAR10/100 eCDF. Figures 2 and 3 present the empirical Cumulative Distribution Functions (eCDFs) of utility calibration errors for models evaluated on CIFAR10 and CIFAR100, respectively, across the broader utility classes \mathcal{U}_{lin} (linear utilities) and $\mathcal{U}_{\text{rank}}$ (rank-based utilities). These plots illustrate the proportion of sampled utility functions for which the calibration error falls below a given threshold. For CIFAR10, most post-hoc methods shift the curves to the left, indicating improved calibration across a wider range of utilities. However, when examining the $\mathcal{U}_{\text{rank}}$ class for RepVGG-CIFAR10, we observe that some post-hoc methods shifted the curves to the right, indicating a degradation of calibration. In addition, observing the $\mathcal{U}_{\text{rank}}$ and \mathcal{U}_{lin} across CIFAR10 results, we observe that post-hoc calibration does not only shift the curve but it also changes its shape. Such more nuanced shifts in calibration performance are not observable in table 2. This suggest that calibration eCDF offers an additional valuable perspective to analyse model calibration. For CIFAR100, RepVGG

1234 followed similar trends as the CIFAR10 experiments. Nonetheless, for ShuffleNet, post-hoc
 1235 methods actively degraded $\mathcal{U}_{\text{rank}}$. Again, this was not observable from Table 3, indicating the added
 1236 value in considering the eCDF plots.

1237 **ImageNet-1K eCDF**. The eCDF plots for the ImageNet-1K models are shown in Figure 4, detailing
 1238 performance across \mathcal{U}_{lin} and $\mathcal{U}_{\text{rank}}$. For both \mathcal{U}_{lin} and $\mathcal{U}_{\text{rank}}$, the uncalibrated model is shifted to
 1239 the right of almost all post-hoc calibrated versions. We also note that post-hoc calibration does
 1240 consistently alter the shape of the curve making it more vertical, indicating a tighter concentration of
 1241 the sampled errors. Similar to CIFAR experiments, the eCDF curves provide an additional perspective
 1242 to contrast and compare the performance of different post-hoc calibration methods.

Table 2: CIFAR10 calibration results, comparing post-hoc methods using Accuracy, Brier Score, binned ECEs (TCE^{Bin} , CWE^{Bin}), and utility calibration errors: \mathcal{U}_{CWE} , \mathcal{U}_{TCE} , and $\mathcal{U}_{\text{topK}}$, with mean \pm maximum deviation over 5 splits.

Method	Accuracy ($\times 10^2$)	Brier Score ($\times 10^2$)	CWE^{bin} ($\times 10^4$)	TCE^{bin} ($\times 10^3$)	\mathcal{U}_{CWE} ($\times 10^4$)	\mathcal{U}_{TCE} ($\times 10^3$)	$\mathcal{U}_{\text{topK}}$ ($\times 10^3$)
Uncalibrated	94.5 \pm 1.30	8.92 \pm 2.03	39.6 \pm 6.71	37.3 \pm 11.7	106 \pm 19.7	39.0 \pm 11.5	39.0 \pm 11.5
Dirichlet	88.0 \pm 0.767	14.4 \pm 0.993	34.9 \pm 18.4	15.2 \pm 12.8	102 \pm 33.3	14.1 \pm 5.91	14.2 \pm 6.00
IR	94.5 \pm 1.30	8.10 \pm 1.55	37.2 \pm 5.85	8.70 \pm 6.00	76.3 \pm 21.6	10.2 \pm 5.02	14.3 \pm 1.91
IR (OvA)	94.6 \pm 1.70	8.12 \pm 1.63	24.9 \pm 5.56	9.05 \pm 8.51	70.4 \pm 16.4	10.9 \pm 8.41	15.7 \pm 2.13
T.S.	94.5 \pm 1.30	8.13 \pm 1.62	62.3 \pm 7.53	21.9 \pm 3.94	87.0 \pm 17.5	18.0 \pm 5.93	23.4 \pm 8.01
Ens.T.S.	94.5 \pm 1.30	8.13 \pm 1.62	62.3 \pm 7.53	21.9 \pm 3.94	87.0 \pm 17.5	18.0 \pm 5.93	23.4 \pm 8.01
V.S.	94.7 \pm 1.37	7.98 \pm 1.72	47.6 \pm 11.6	17.0 \pm 6.02	87.6 \pm 29.7	14.6 \pm 2.96	16.4 \pm 5.30
Patch	94.7 \pm 1.23	8.03 \pm 1.60	38.1 \pm 13.0	9.09 \pm 9.68	77.3 \pm 15.0	10.1 \pm 3.05	11.1 \pm 3.05

RepVGG-CIFAR10

Method	Accuracy ($\times 10^2$)	Brier Score ($\times 10^2$)	CWE^{bin} ($\times 10^4$)	TCE^{bin} ($\times 10^3$)	\mathcal{U}_{CWE} ($\times 10^4$)	\mathcal{U}_{TCE} ($\times 10^3$)	$\mathcal{U}_{\text{topK}}$ ($\times 10^3$)
Uncalibrated	93.4 \pm 1.17	10.8 \pm 2.20	55.4 \pm 16.2	42.6 \pm 11.6	129 \pm 32.5	42.8 \pm 14.1	42.8 \pm 14.1
Dirichlet	83.5 \pm 1.57	21.3 \pm 0.802	51.5 \pm 16.0	22.4 \pm 15.4	145 \pm 50.6	22.1 \pm 18.5	22.9 \pm 17.9
IR	93.4 \pm 1.17	9.89 \pm 1.58	50.4 \pm 11.4	11.7 \pm 7.01	97.4 \pm 23.6	11.5 \pm 6.41	15.2 \pm 2.79
IR (OvA)	93.6 \pm 1.17	9.81 \pm 1.67	33.8 \pm 11.5	11.4 \pm 2.50	88.8 \pm 20.5	10.6 \pm 6.18	18.0 \pm 2.57
T.S.	93.4 \pm 1.17	9.89 \pm 1.69	71.5 \pm 13.6	18.4 \pm 1.52	105 \pm 27.2	17.9 \pm 6.94	22.4 \pm 9.51
Ens.T.S.	93.4 \pm 1.17	9.89 \pm 1.69	71.5 \pm 13.6	18.4 \pm 1.52	105 \pm 27.2	17.9 \pm 6.94	22.4 \pm 9.51
V.S.	93.6 \pm 1.27	9.61 \pm 1.68	47.2 \pm 9.17	13.6 \pm 4.60	99.0 \pm 50.3	12.7 \pm 6.32	14.9 \pm 4.32
Patch	93.5 \pm 1.17	9.80 \pm 1.64	47.9 \pm 19.0	11.4 \pm 8.61	94.2 \pm 15.9	12.3 \pm 10.2	12.6 \pm 9.61

ShuffleNet-CIFAR10

Table 3: CIFAR100 calibration results, comparing post-hoc methods using Accuracy, Brier Score, binned ECEs (TCE^{Bin} , CWE^{Bin}), and utility calibration errors: \mathcal{U}_{CWE} , \mathcal{U}_{TCE} , and $\mathcal{U}_{\text{topK}}$, with mean \pm maximum deviation over 5 splits.

Method	Accuracy ($\times 10^2$)	Brier Score ($\times 10^2$)	CWE^{bin} ($\times 10^4$)	TCE^{bin} ($\times 10^3$)	\mathcal{U}_{CWE} ($\times 10^4$)	\mathcal{U}_{TCE} ($\times 10^3$)	$\mathcal{U}_{\text{topK}}$ ($\times 10^3$)
Uncalibrated	77.2 \pm 1.47	32.7 \pm 2.55	15.2 \pm 1.75	58.1 \pm 22.1	72.8 \pm 14.6	55.5 \pm 2.25	55.6 \pm 2.43
Dirichlet	71.6 \pm 4.60	40.9 \pm 16.3	20.8 \pm 4.21	61.0 \pm 221	86.6 \pm 38.9	61.8 \pm 219	65.5 \pm 216
IR	77.2 \pm 1.47	31.9 \pm 2.37	14.1 \pm 1.55	20.1 \pm 11.3	65.6 \pm 11.2	15.4 \pm 4.59	46.6 \pm 8.74
IR (OvA)	76.6 \pm 1.73	32.8 \pm 3.29	14.4 \pm 1.60	30.4 \pm 13.0	65.3 \pm 11.4	33.2 \pm 11.7	50.0 \pm 4.75
T.S.	77.2 \pm 1.47	32.5 \pm 2.50	17.0 \pm 1.52	45.4 \pm 22.2	72.6 \pm 15.6	37.2 \pm 4.80	66.0 \pm 4.91
Ens.T.S.	77.2 \pm 1.47	32.4 \pm 2.46	20.0 \pm 0.965	40.4 \pm 18.2	73.4 \pm 15.2	27.2 \pm 6.50	85.8 \pm 6.30
V.S.	76.8 \pm 2.17	33.0 \pm 3.17	17.8 \pm 1.94	50.5 \pm 13.3	72.9 \pm 11.4	45.4 \pm 11.1	52.2 \pm 5.74
Patch	77.1 \pm 1.63	32.1 \pm 2.04	20.6 \pm 5.05	26.9 \pm 14.4	66.8 \pm 6.25	21.6 \pm 7.71	36.7 \pm 21.6

RepVGG-CIFAR100

Method	Accuracy ($\times 10^2$)	Brier Score ($\times 10^2$)	CWE^{bin} ($\times 10^4$)	TCE^{bin} ($\times 10^3$)	\mathcal{U}_{CWE} ($\times 10^4$)	\mathcal{U}_{TCE} ($\times 10^3$)	$\mathcal{U}_{\text{topK}}$ ($\times 10^3$)
Uncalibrated	75.4 \pm 2.47	35.2 \pm 3.48	13.5 \pm 2.41	76.1 \pm 19.2	74.2 \pm 16.0	80.2 \pm 22.9	80.2 \pm 22.9
Dirichlet	65.0 \pm 7.17	49.7 \pm 23.4	25.3 \pm 4.35	86.2 \pm 279	128 \pm 23.9	85.9 \pm 284	101 \pm 265
IR	75.4 \pm 2.47	34.2 \pm 3.06	14.3 \pm 2.22	17.3 \pm 13.2	68.0 \pm 10.5	19.0 \pm 9.16	49.4 \pm 5.77
IR (OvA)	75.2 \pm 2.80	35.2 \pm 3.59	14.5 \pm 2.91	35.0 \pm 17.1	71.8 \pm 15.4	34.3 \pm 21.4	54.8 \pm 2.94
T.S.	75.4 \pm 2.47	34.6 \pm 3.09	16.4 \pm 1.82	37.6 \pm 17.0	72.9 \pm 13.5	32.3 \pm 19.3	54.3 \pm 4.63
Ens.T.S.	75.4 \pm 2.47	34.5 \pm 3.12	19.0 \pm 1.89	31.4 \pm 20.9	73.8 \pm 14.2	25.0 \pm 17.9	70.3 \pm 7.57
V.S.	75.0 \pm 2.90	34.9 \pm 3.67	17.9 \pm 2.80	40.8 \pm 24.0	77.5 \pm 20.9	39.2 \pm 20.3	48.5 \pm 6.36
Patch	75.2 \pm 2.67	34.6 \pm 3.21	22.8 \pm 6.14	35.4 \pm 29.3	76.2 \pm 33.3	28.5 \pm 18.3	47.7 \pm 38.1

ShuffleNet-CIFAR100

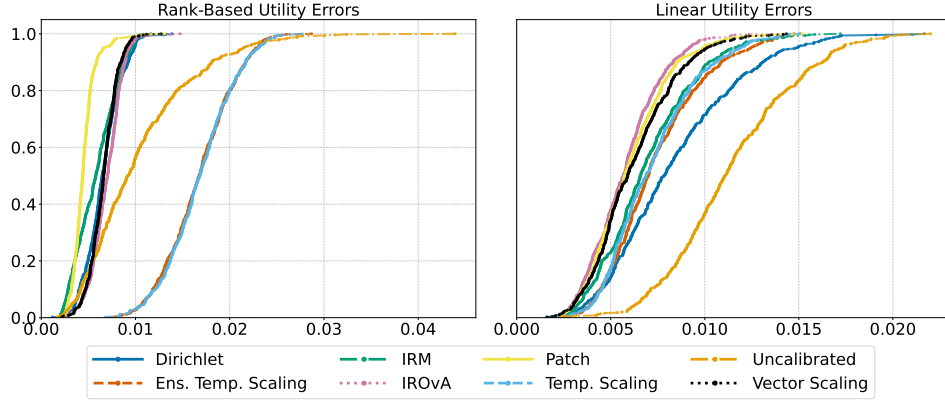
Table 4: ImageNet calibration results, comparing post-hoc methods using Accuracy, Brier Score, binned ECEs (TCE^{Bin} , CWE^{Bin}), and utility calibration errors: \mathcal{U}_{CWE} , \mathcal{U}_{TCE} , and $\mathcal{U}_{\text{topK}}$, with mean \pm maximum deviation over 5 splits.

Method	Accuracy ($\times 10^2$)	Brier Score ($\times 10^2$)	CWE^{bin} ($\times 10^4$)	TCE^{bin} ($\times 10^3$)	\mathcal{U}_{CWE} ($\times 10^4$)	\mathcal{U}_{TCE} ($\times 10^3$)	$\mathcal{U}_{\text{topK}}$ ($\times 10^3$)
Uncalibrated	77.8 \pm 1.11	32.2 \pm 1.26	2.13 \pm 0.0790	69.6 \pm 7.57	33.1 \pm 5.24	71.7 \pm 12.1	95.1 \pm 3.50
Dirichlet	73.7 \pm 1.02	36.8 \pm 1.18	1.79 \pm 0.184	14.3 \pm 15.2	34.0 \pm 5.03	23.5 \pm 14.3	30.6 \pm 2.60
IR	77.8 \pm 1.11	31.6 \pm 1.34	1.44 \pm 0.0991	13.3 \pm 4.49	32.0 \pm 3.24	19.1 \pm 2.04	31.5 \pm 2.59
IR (OvA)	77.1 \pm 0.867	33.1 \pm 1.61	1.43 \pm 0.0714	41.5 \pm 13.7	31.2 \pm 1.45	38.9 \pm 11.9	68.2 \pm 7.41
T.S.	77.8 \pm 1.11	31.4 \pm 1.36	1.61 \pm 0.0975	13.8 \pm 4.84	32.6 \pm 4.17	19.5 \pm 5.77	32.1 \pm 5.48
Ens.T.S.	77.8 \pm 1.11	31.4 \pm 1.36	1.66 \pm 0.0845	12.9 \pm 5.84	32.6 \pm 4.24	19.0 \pm 3.91	36.5 \pm 7.83
V.S.	77.5 \pm 1.19	31.6 \pm 1.39	1.52 \pm 0.0562	29.6 \pm 9.42	31.2 \pm 4.62	32.6 \pm 10.6	32.8 \pm 9.27
Patch	77.7 \pm 1.11	32.4 \pm 1.17	3.10 \pm 0.346	45.8 \pm 21.3	40.7 \pm 23.2	46.1 \pm 14.1	71.5 \pm 36.6

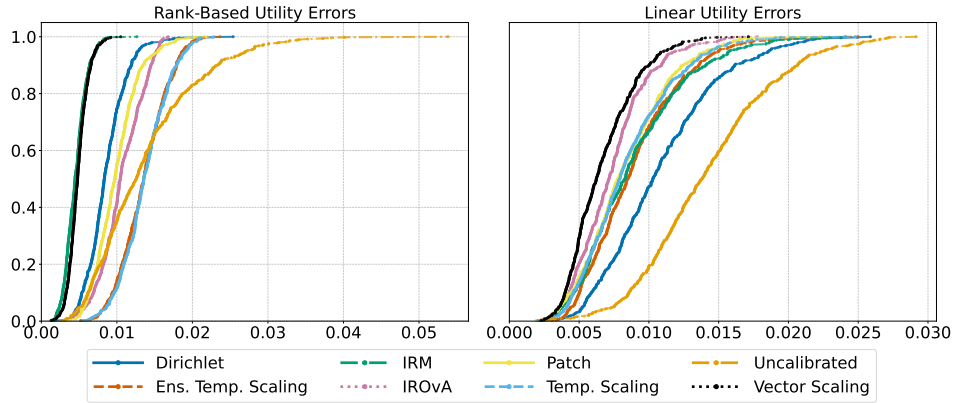
EfficientNet-ImageNet

Method	Accuracy ($\times 10^2$)	Brier Score ($\times 10^2$)	CWE^{bin} ($\times 10^4$)	TCE^{bin} ($\times 10^3$)	\mathcal{U}_{CWE} ($\times 10^4$)	\mathcal{U}_{TCE} ($\times 10^3$)	$\mathcal{U}_{\text{topK}}$ ($\times 10^3$)
Uncalibrated	85.2 \pm 0.880	22.6 \pm 0.965	2.46 \pm 0.0635	94.2 \pm 7.10	30.4 \pm 2.70	94.8 \pm 7.88	124 \pm 2.71
Dirichlet	84.6 \pm 0.607	22.6 \pm 1.04	1.30 \pm 0.0731	15.2 \pm 4.95	28.7 \pm 2.63	19.5 \pm 7.53	26.0 \pm 3.44
IR	85.2 \pm 0.880	21.6 \pm 1.05	1.10 \pm 0.0809	7.49 \pm 4.58	28.7 \pm 1.81	15.6 \pm 5.47	19.6 \pm 3.66
IR (OvA)	84.9 \pm 0.800	22.9 \pm 1.09	1.10 \pm 0.0562	33.1 \pm 4.72	28.0 \pm 2.74	33.0 \pm 3.28	55.6 \pm 4.90
T.S.	85.2 \pm 0.880	21.6 \pm 1.10	1.50 \pm 0.0447	21.2 \pm 4.89	29.3 \pm 1.61	21.2 \pm 5.75	45.2 \pm 4.15
Ens.T.S.	85.2 \pm 0.880	21.6 \pm 1.10	1.50 \pm 0.0447	21.2 \pm 4.89	29.3 \pm 1.61	21.2 \pm 5.75	45.2 \pm 4.15
V.S.	85.2 \pm 0.653	21.9 \pm 1.12	1.22 \pm 0.0895	32.1 \pm 6.21	27.3 \pm 2.29	35.4 \pm 8.70	35.4 \pm 8.70
Patch	85.2 \pm 0.900	21.8 \pm 1.21	1.87 \pm 0.373	19.2 \pm 9.35	34.7 \pm 28.9	21.0 \pm 10.8	30.2 \pm 18.0

ViT_Base_P16_224-ImageNet

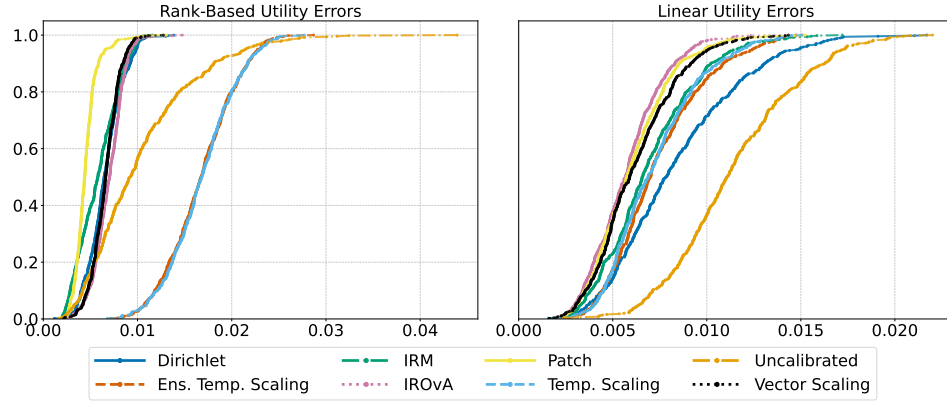


RepVGG-CIFAR10

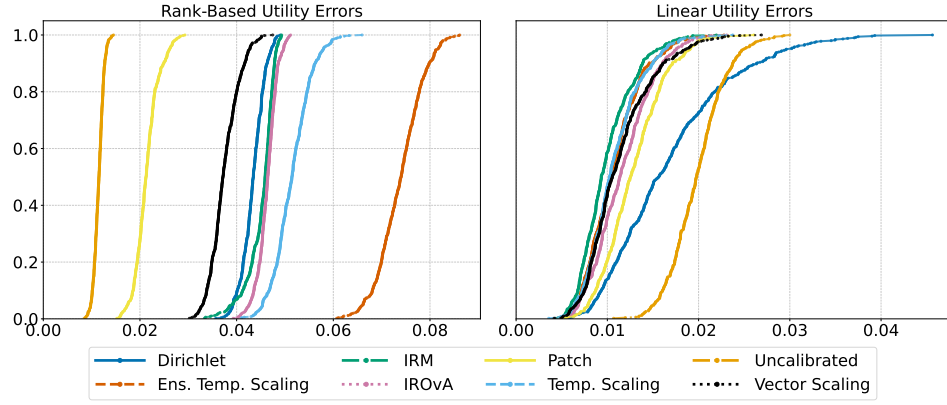


ShuffleNet-CIFAR10

Figure 2: CIFAR10 eCDF plots for \mathcal{U}_{lin} and $\mathcal{U}_{\text{rank}}$.

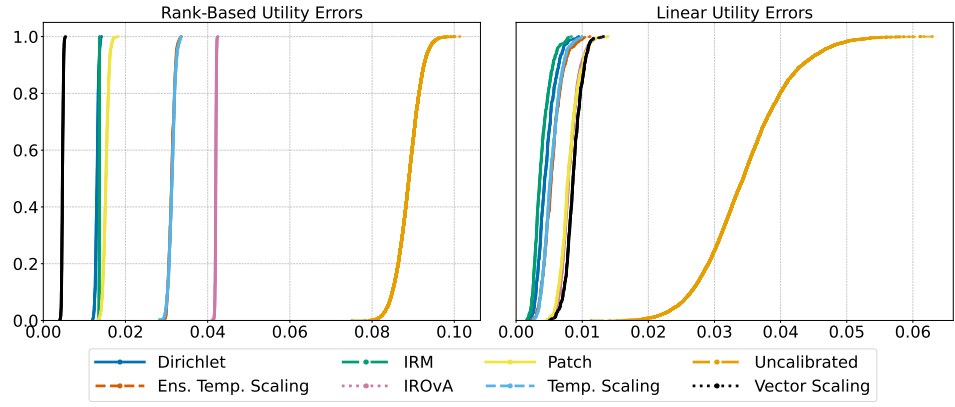


RepVGG-CIFAR100

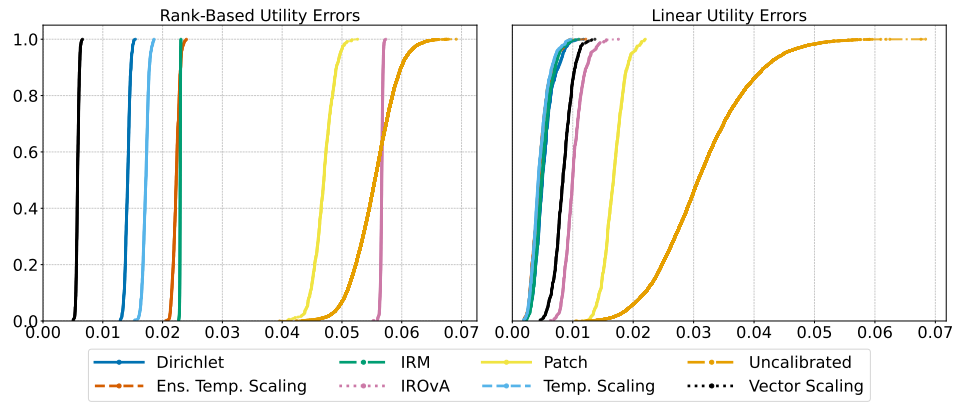


ShuffleNet-CIFAR100

Figure 3: CIFAR100 eCDF plots for \mathcal{U}_{lin} and $\mathcal{U}_{\text{rank}}$.



ViT_Base_P16_224-ImageNet1k



EfficientNet-ImageNet1k

Figure 4: ImageNet eCDF plots for \mathcal{U}_{lin} and $\mathcal{U}_{\text{rank}}$.