Scalable Utility-Aware Multiclass Calibration

Mahmoud Hegazy
CMAP, École polytechnique,
IP Paris, France;
Inria Paris, France

Michael I. Jordan
Inria Paris, France;
EECS, UC Berkeley, USA

Aymeric Dieuleveut
CMAP, École polytechnique,
IP Paris, France

Abstract

Ensuring that classifiers are well-calibrated, i.e., their predictions align with observed frequencies, is a minimal and fundamental requirement for classifiers to be viewed as trustworthy. Existing methods for assessing multiclass calibration often focus on specific aspects associated with prediction (e.g., top-class confidence, class-wise calibration) or utilize computationally challenging variational formulations. In this work, we study scalable *evaluation* of multiclass calibration. To this end, we propose utility calibration, a general framework that measures the calibration error relative to a specific utility function that encapsulates the goals or decision criteria relevant to the end user. We demonstrate how this framework can unify and re-interpret several existing calibration metrics, particularly allowing for more robust versions of the top-class and class-wise calibration metrics, and, going beyond such binarized approaches, toward assessing calibration for richer classes of downstream utilities.

1 INTRODUCTION

Calibration is a fundamental property of probabilistic predictors. A calibrated model produces predictions that, on average, align with observed frequencies. For instance, if a weather forecaster predicts a 30% chance of rain on a given day, rain should occur on approximately 30% of such days. In multiclass classification problems, calibration ensures that the predicted probabilities reflect the true likelihood of each class. Formally, let $\mathcal{X}$ denote the input space, $\mathcal{Y} = \{e_1, \dots, e_C\}$ the output space, where e_i is the i-th canonical basis vector in $\mathbb{R}^C$, and $\Delta^{C-1} := \left\{x \in \mathbb{R}^C_+ | \sum_i x_i = 1\right\}$ denote the simplex in $\mathbb{R}^C$. A predictor $f : \mathcal{X} \to \Delta^{C-1}$ is said to be perfectly calibrated with respect to 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 deviation from perfect calibration is the Mean Calibration Error (MCE) (Vaicenavicius et al., 2019).

Definition 1.1 (Mean Calibration Error)**.** *For a distribution D such that $(X, Y) \sim D$ and a predictor f, the mean calibration error is defined as*

$$\mathrm{MCE}(f) := \mathbb{E}\left[\|\mathbb{E}[Y \mid f(X)] - f(X)\|_2^2\right].$$

Without further assumptions, the MCE is fundamentally impossible to estimate, even in the binary setting (Lee et al., 2023; Duchi, 2024). While assumptions like Hölder continuity of $\mathbb{E}[Y|f(X)]$ allow for consistent estimators of $\mathbb{E}[Y|f(X)]$ or minimax optimal tests for $\mathrm{MCE}(f)$ (Lee et al., 2023; Popordanoska et al., 2022; Tsybakov, 2009), their sample complexity scales exponentially with the dimension C.

Due to the difficulty of measuring MCE, multiple relaxations have been proposed, falling into two main categories: *binarized* and *variational*. First, binarized approaches (Gupta and Ramdas, 2022; Panchenko et al., 2022; Guo et al., 2017) simplify the problem by focusing on specific binary events derived from the multiclass predictions, e.g. top-class or class-wise calibration. However, these methods are by nature presumptive of downstream tasks. Moreover, their reliance on binning schemes or kernel estimators for the underlying binary subproblems introduces sensitivity to estimator choices and can suffer from high bias (Roelofs et al., 2022). Second, variational approaches (Jung et al., 2021; Błasiok et al., 2023; Gopalan et al., 2024; Kumar et al., 2018; Widmann et al., 2019; Zhao et al., 2021) assess calibration through optimization problems, such as the distance to the nearest perfectly calibrated predictor or the worst-case error against a class of witness functions.

Proceedings of the 29th International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300.

Unfortunately, these methods can be computationally intensive and can scale poorly in C.

To address these limitations and provide an application-focused perspective on calibration, we introduce *utility calibration*. This framework evaluates a model f by considering a downstream user who employs its predictions $f(X)$. The core idea is to measure calibration error relative to a specific *utility function*, denoted u, which encapsulates the goals, costs, or decision criteria relevant to this end user. Utility calibration then assesses how well the *expected utility* (as estimated by the user based on $f(X)$ and u) aligns with the *realized utility* (obtained when the true outcome Y is observed). For example, in medical diagnosis, a classifier may output probabilities over C diseases, with the relative severity of each disease encoded through a cost vector $a \in [-1, 1]^C$. A clinician may be interested in $\langle f(X), a\rangle$, the model's predicted expected risk. In this context, utility calibration aims to assess if $\langle f(X), a\rangle$ is a calibrated proxy for the true realized cost $\langle a, Y\rangle$.

In practice, models often serve diverse users or a single user with multiple objectives (Elkan, 2001). Utility calibration naturally extends to handle *classes of utility functions*: the utility calibration error for a class $\mathcal{U}$ is defined as the worst-case error over $u \in \mathcal{U}$, denoted $\mathrm{UC}(f, \mathcal{U})$. A notable aspect of this formulation is that it provides a structured way to express and analyze various existing calibration notions. In particular, by defining appropriate utility functions within $\mathcal{U}$, concepts such as top-class and class-wise calibration can be cast within the utility calibration framework. This offers a unified perspective and a superior alternative to binning for examining those notions of calibration.

Contributions. We introduce **utility calibration** (Section 3), a unified, binning-free framework that generalizes standard metrics such as top-class and class-wise calibration to arbitrary utility functions, recovering and refining them as special cases. The induced metrics upper-bound commonly used binned variants up to a factor equal to the number of bins. We further show that the framework inherits **decision-theoretic guarantees** analogous to those of Rossellini et al. (2025), where a small utility calibration error bounds the gain from monotone post-processing of the predicted utility. On the computational side (Section 4), we distinguish *proactive* from *interactive* measurability. While proactively finding the worst-case utility is hard for expressive classes, interactive estimation is achievable with sample complexity $\tilde{O}(C)$, uniformly over any fixed u. We show that standard patching algorithms from the multicalibration literature apply directly (Section 4.1). Finally, experiments on ImageNet-1K (Section 5) validate our framework, demonstrating eCDF-based evaluation across utility classes.

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

2 RELATED WORK

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

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

Popular post-hoc calibration algorithms include Temperature Scaling and its multi-parameter extensions, Vector Scaling and Matrix Scaling (Guo et al., 2017), which may all be regarded as a multiclass extension of Platt's scaling (Platt et al., 1999). Dirichlet calibration assumes the model's predicted probability vectors can be modeled by a Dirichlet distribution, whose parameters are learned on a calibration set to transform the original probabilities (Kull et al., 2019). Nonparametric methods such as Histogram Binning (Zadrozny and Elkan, 2001) and Isotonic Regression (Zadrozny and Elkan, 2002) learn calibration maps by discretizing the probability space or fitting monotonic (order-preserving) functions, respectively. Other methods also include the approaches in: Patel et al. (2020), which applies a specific binning strategy followed by recalibration to minimize class-wise calibration error, Rahimi et al. (2020), which uses order-preserving transformations for recalibration to maintain accuracy. Finally, a related body of literature aims to improve calibration by regularizing the training objective, e.g. (Mukhoti et al., 2020; Popordanoska et al., 2022; Marx et al., 2024).

While post-hoc methods aim to minimize the calibration error, assessing the calibration error itself remains a non-trivial task. **Binarized relaxations** aim to circumvent the difficulty of measuring the calibration error of a high-dimensional predictor f by measuring the MCE of a single or multiple downstream binary versions of f instead. Two commonly used relaxations

are the Top-Class calibration Error (TCE) (Guo et al., 2017) and the Class-Wise calibration Error (CWE) (Panchenko et al., 2022), defined as

$$\mathrm{TCE}(f) = \mathbb{E}\left[\,|\,\mathbb{E}[Y_* \mid p_*] - p_*\,|\,\right], \quad \text{and}$$
$$\mathrm{CWE}(f) = \sum_{i\in[C]} w_i\,\mathbb{E}\left[\,|\,\mathbb{E}[\mathbf{1}_{Y=e_i} \mid f_i] - f_i\,|\,\right],$$

with the shorthands $f_i \coloneqq f(X)_i$, $p_* \coloneqq f(X)_{i_*}$, and $Y_* \coloneqq \mathbf{1}_{Y=e_{i_*}}$, where $i_* \coloneqq \mathrm{argmax}_i f(X)_i$. In addition, w_i is a class-dependent weight, which can be set to $1/C$, $w_i = \mathbb{P}\{Y = e_i\}$, or another choice $w \in \Delta^{C-1}$.

Both TCE and CWE require the estimation of conditional expectation, which is typically approximated using binning schemes. For $(B_j)_{j\in[m]}$ a partition of $[0,1]$, the binned estimators are

$$\mathrm{TCE}^{\mathrm{bin}}(f) = \sum_{j\in[m]} \left|\,\mathbb{E}\left[(p_* - Y_*)\mathbf{1}_{p_*\in B_j}\right]\,\right| \tag{2.1}$$

$$\mathrm{CWE}^{\mathrm{bin}}(f) = \sum_{i\in[C],j\in[m]} w_i \left|\mathbb{E}\left[\left(f_i - \mathbf{1}_{Y=e_i}\right)\mathbf{1}_{f_i\in B_j}\right]\right|. \tag{2.2}$$

Gupta and Ramdas (2022) unified multiple instances of binarized proxies of MCE, such as TCE, CWE and topK confidence calibration, introduced in (Gupta et al., 2021), 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 (Roelofs et al., 2022; Gruber and Buettner, 2022). Apart from the simpler equal-size bins (Guo et al., 2017) and equal-weight bins (Zadrozny and Elkan, 2001), multiple binning schemes built on top of different heuristics have been proposed (see, e.g., Roelofs et al., 2022; Patel et al., 2020; Naeini et al., 2015; Nixon et al.). Gupta and Ramdas (2021) showed a simple equal-weight binning scheme with better sample complexity guarantees for estimating bin averages. Kumar et al. (2019) 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 $\mathrm{TCE}^{\mathrm{bin}}(f)$ as an estimate of $\mathrm{TCE}(f)$ is lower bounded by 0.49 (noting that TCE is bounded between 0 and 1).

Meanwhile, there exist binning-free alternatives for binarized reductions (see, e.g., Popordanoska et al., 2022; Gupta et al., 2021). Nonetheless, in an assumption-free setting, it is generally impossible to consistently estimate the MCE of binary predictors (Lee et al., 2023; Duchi, 2024; Rossellini et al., 2025). As such, it is generally difficult to control the calibration error defined by binarized relaxations.

On the other hand, **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) measures the distance between f and the nearest perfectly calibrated predictors (Błasiok et al., 2023):

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

A unified formulation of variational measures of calibration is weighted calibration, which assesses the calibration error against a class of witness functions (Jung et al., 2021). 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

$$\mathrm{CE}_{\mathcal{W}}(f) = \sup_{w\in\mathcal{W}} \mathbb{E}_{X,Y}\left[\big\langle w\big(f(X)\big), f(X) - Y\big\rangle\right]. \tag{2.3}$$

A specific instance of weighted calibration is the Kernel Calibration Error (KCE) (Widmann et al., 2019; Lin et al., 2023), 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. (2023) showed that in the binary setting, $\mathrm{DC}(f)$ and $\mathrm{CE}_{\mathrm{Lip}(1)}(f)$ are equivalent up to a (low-degree) polynomial scaling and that $\mathrm{CE}_{\mathrm{Lip}(1)}(f)$ can be well approximated by the RKHS of the Laplace kernel, where $\mathrm{Lip}(1)$ is the class of 1-Lipschitz functions from Δ^{C-1} to $[-1,1]$.

The result on the equivalence between $\mathrm{CE}_{\mathrm{Lip}(1)}(f)$ and $\mathrm{DC}(f)$ was further extended to the multiclass setting in (Duchi, 2024, Theorem 15.5.5) and (Gopalan et al., 2024, Lemma 3.3). In particular, Gopalan et al. (2022a) showed that measuring either $\mathrm{DC}(f)$ or $\mathrm{CE}_{\mathrm{Lip}(1)}(f)$ requires an exponential number of samples in C (Gopalan et al., 2024, Theorem 3.2. and Theorem 3.4.). Thus, even though $\mathrm{DC}\,(f)$ can be efficiently assessed in the binary setting, it is quickly intractable as C increases.

A particular case is *Decision calibration*, introduced by Zhao et al. (2021), 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:

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

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 actions when using the predictor. A key contribution of Zhao et al. (2021) is showing that decision calibration of order K can be achieved by having $\sup_{p\in P(K)} \|\mathbb{E}[(Y - f(X))\mathbf{1}\{f(X)\in p\}]\| = 0$, where $P(K)$ is the set of polytopes with at most K supporting hyperplanes. Moreover, low decision calibration guarantees a notion of no-regret to downstream users, i.e., they cannot improve their utility by using any other best response policy; associated with another loss ℓ' instead of ℓ.

With a similar aim to Zhao et al. (2021), multiple works have studied calibration through a downstream decision-theoretic perspective. Broadly, two complementary approaches have emerged: (i) online, regret-driven forecasting that enforces event-conditional notions of calibration and yields no (swap)-regret guarantees for best-responding agents (Noarov et al., 2025; Roth and Shi, 2024); and (ii) decision-centric objectives that guarantee good performance across rich families of utilities or proper scoring rules, e.g., U-calibration (Kleinberg et al., 2023).

Foundationally, these formulations target an ambitious goal, ensuring that downstream agents who best-respond using the predictor f achieve (near-)optimal utility across scenarios and losses. However, this ambition often comes with scalability costs: U-calibration is developed in the binary online setting; the swap-regret guarantees in Roth and Shi (2024) rely on Lipschitz utilities in low dimensions (e.g., $d \in \{1, 2\}$) and, in higher dimensions, assume a bounded action set and smooth best responses; and measuring decision calibration is computationally intractable in C, with exponential complexity even for $K = 2$ (Gopalan et al., 2024).

By contrast, we adopt a scalable assessment perspective. Our utility calibration (shortly introduced in Section 3) requires that the predicted utility $v_u(X)$ is a reliable regressor of the realized utility $u(f(X), Y)$. This prioritizes the reliability of utility estimation–not enforcing optimal downstream decisions–so users can recognize when their expected utility is poor rather than be misled by optimistic forecasts. Operationally, it reduces multiclass assessment to a binning-free worst-interval deviation in v_u, scales well with C, and supports interactively measurable audits across broad utility classes, complementing online/regret-focused formulations.

3 UTILITY CALIBRATION (UC)

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

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

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

$$\sup_{I\in\mathbb{I}[-1,1]} \left|\mathbb{E}\left[(u(f(X), Y) - v_u(X))\mathbf{1}_{v_u(X)\in I}\right]\right|. \quad (3.2)$$

We say that f is ε-calibrated with respect to a utility function u if $\mathrm{UC}(f, u) \le \varepsilon$. Note that for $I = [a, b]$, the inner term in (3.2) can be rewritten as

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

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

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

$$\sup_{I\in\mathbb{I}[-1,1]} \left|\mathbb{E}[\langle Y - f(X), \vec{u}(X)\rangle\, \mathbf{1}\{v_u(X) \in I\}]\right|. \quad (3.3)$$

Thus, utility calibration is equivalent to weighted calibration (2.3) with the witness class $\mathcal{W}(u) := \{x \mapsto \xi\vec{u}(x)\mathbf{1}\{v_u(x) \in I\} \mid I \in \mathbb{I}[-1, 1], \xi \in \{-1, 1\}\}$. This equivalence shows that utility calibration is exactly a weighted calibration problem with an interpretable, application-driven witness class. It also unlocks the toolkit of weighted calibration algorithms.

Remark 3.1 (Connection to Outcome Indistinguishability)**.** *Utility calibration requires that $\hat{Y} \sim f(X)$ yields an unbiased estimate of the utility, which structurally aligns with Outcome Indistinguishability (OI) (Dwork et al., 2021): a predictor is reliable when simulated outcomes $\hat{Y} \sim f(X)$ are indistinguishable from Nature's*

true outcomes Y. This perspective connects to recent work leveraging OI variants to link loss minimization, omnipredictors, and multicalibration (Gopalan et al., 2022b, 2023a,b). We also note that the entire treatment carries over to losses $l = -u$.

3.1 Decision-Theoretic Implications of UC

Utility calibration assesses $\mathrm{UC}(f, u)$ via the worst-case interval of $v_u(\cdot)$, which generalizes the binary Cut-Off calibration of Rossellini et al. (2025) to the multiclass setting and arbitrary utility functions. The decision-theoretic guarantees below (Propositions 3.2 and 3.3) extend those of Rossellini et al. (2025, Props. 2.1 and 3.2) accordingly.

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 $\hat{U}_{t_0} := \mathbf{1}\{v_u(X) \geq t_0\}$. 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 $\ell_{\text{util}}(\tilde{u}, \widehat{U}; t) = |\tilde{u} - t|\, \mathbf{1}\{\widehat{U} \neq \mathbf{1}\{u \geq t\}\}$, 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 $v_u(\cdot)$ with a monotone function.

Proposition 3.2 (Utility Risk Gap). *For any utility u and threshold $t_0 \in [-1, 1]$,*

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

In words, Proposition 3.2 shows that the risk gap from any monotone post-processing of v_u is bounded by $2\,\mathrm{UC}(f, u)$: when f is well utility-calibrated, no such post-processing can yield a meaningful improvement. 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. (2025, 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 $\mathbb{E}[u_Y \mid g_u(X)] = g_u(X)$, a.s. We denote this extended notion of distance from calibration as $\mathrm{DCU}(f, u)$, the Distance to Calibrated Utility Predictor for $v_u(X)$ with respect to the realized utility $u(f(X), Y)$:

$$\mathrm{DCU}(f) := \inf_{g_u:\mathcal{X}\to[-1,1]} \mathbb{E}\,|g_u(X) - v_u(X)| \quad \text{s.t.} \quad \mathbb{E}[u_Y \mid g_u(X)] = g_u(X)\,.$$

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

Proposition 3.3 (Utility Calibration Upper Bounds DCU). *Let $u : \Delta^{C-1} \times \mathcal{Y} \to [-1, 1]$ be a utility function. Then,*

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

Proposition 3.3 implies that if $\mathrm{UC}(f, u)$ 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 $\mathrm{UC}(f, u)$: not only does it *ensure actionable decisions based on $v_u(X)$*, but it also *guarantees that $v_u(X)$ is not far from calibration.*

3.2 Measuring $\mathrm{UC}(f, u)$

A naturally arising question is on the difficulty of measuring and achieving a small utility calibration error. We show in Lemma 3.4 that both the computational and sample complexity of estimating $\mathrm{UC}(f, u)$ are generally feasible and of limited dependence on the dimension, allowing its scalability to predictors with thousands of classes.

Lemma 3.4 (Estimating Utility Calibration Against a Single Function). *Let $u : \Delta^{C-1} \times \mathcal{Y} \to [-1, 1]$ be a fixed utility function and $f : \mathcal{X} \to \Delta^{C-1}$ be a given predictor. Define the empirical estimator $\widehat{\mathrm{UC}}(f, u; S)$ based on n i.i.d. samples $S = \{(X_i, Y_i)\}_{i=1}^n \sim D^n$ as*

$$\sup_{I\in\mathbb{I}[-1,1]} \left| \frac{1}{n} \sum_{i=1}^{n} \left[(u(f(X_i), Y_i) - v_u(X_i)) \mathbf{1}_{v_u(X_i)\in I} \right] \right|.$$

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

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

Furthermore, if $v_u(X)$ has a continuous distribution, then $\widehat{\mathrm{UC}}(f, u; S)$ can be computed from S in $O(n \log(n) + nT_{eval})$ time, where T_{eval} is the time to evaluate $f(X_i)$ and $u(\cdot, \cdot)$.

First, we note that the constants hidden in the $\tilde{O}(\cdot)$ in (3.4) are dimension-independent. Similarly, the only dimension-dependent term in the computational complexity is T_{eval}. In particular, $\widehat{\mathrm{UC}}(f, u; S)$ is computed by sorting the n samples by predicted utility ($O(n \log n)$) and then running a linear scan (via Kadane's algorithm) to find the maximum-deviation interval, yielding the $O(n \log n + nT_{eval})$ computational complexity. As such, $\mathrm{UC}(f, u)$ is a completely scalable notion of calibration, allowing it to be implemented for classifiers with a thousand classes. With these facts on the utility calibration w.r.t. a single u, we next turn our attention to Utility Calibration against a class $\mathcal{U}$.

3.3 Utility calibration against a function class

In many real-world scenarios, a single probabilistic predictor f serves multiple downstream users, or a single user employs it under varying conditions or objectives. The exact utility function relevant at the time of decision-making may not be known beforehand, or it may change over time (e.g., due to changing costs, available actions, or strategic goals), or it may be fundamentally user-dependent.

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

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

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

Example 3.5 (Top-Class and Class-Wise Utilities ($\mathcal{U}_{\mathrm{TCE}},\mathcal{U}_{\mathrm{CWE}}$))**.** *Again using the shorthands* $f_i \coloneqq f(X)_i$, $p_* \coloneqq f(X)_{i_*}$, *and* $Y_* \coloneqq \mathbf{1}_{Y=e_{i_*}}$, *where* $i_* \coloneqq \mathrm{argmax}_i f(X)_i$, *define*

$$\mathrm{UC}(f,\mathcal{U}_{\mathrm{TCE}}) = \sup_{I\subseteq[0,1]} |\,\mathbb{E}[(Y_* - p_*)\mathbf{1}_{p_*\in I}]|,$$
$$\mathrm{UC}(f,\mathcal{U}_{\mathrm{CWE}}) = \sup_{c\in[C],I\subseteq[0,1]} |\,\mathbb{E}[(\mathbf{1}_{Y=e_c} - f(X)_c)\mathbf{1}_{f(X)_c\in I}]|.$$

A key observation is that $\mathcal{U}_{\mathrm{TCE}}$ and $\mathcal{U}_{\mathrm{CWE}}$ are not fundamental properties of f. In particular, these conventional notions encode *implicit assumptions* about how the predictor will be used. Top-class calibration presumes that only the maximum probability class matters to the user; class-wise calibration presumes each class is evaluated in isolation (one-vs-all). The utility calibration framework makes these assumptions explicit as each metric becomes a named utility class, rendering implicit assumptions explicit.

In contrast to the binned estimators $\mathrm{TCE}^{\mathrm{bin}}$ (2.1) and $\mathrm{CWE}^{\mathrm{bin}}$ (2.2), utility calibration with $\mathcal{U}_{\mathrm{TCE}}$ and $\mathcal{U}_{\mathrm{CWE}}$ offers a more robust, binning-free assessment: $\mathrm{UC}(f,\mathcal{U}_{\mathrm{TCE}})$ and $\mathrm{UC}(f,\mathcal{U}_{\mathrm{CWE}})$ maximize the calibration deviation over *any* interval $I\subseteq[0,1]$, identifying the worst-case deviation rather than fixing a binning heuristic. This avoids pathologies where bin choices drastically alter estimated errors (Roelofs et al., 2022; Kumar et al., 2019). Consequently, for any m-bin scheme, $m\cdot\mathrm{UC}(f,\mathcal{U}_{\mathrm{TCE}})$ and $m\cdot\mathrm{UC}(f,\mathcal{U}_{\mathrm{CWE}})$ upper bound $\mathrm{TCE}^{\mathrm{bin}}(f)$ and $\mathrm{CWE}^{\mathrm{bin}}(f)$ respectively, while the converse does not hold. We refer to Appendix B.2 for the formal statement. Furthermore, by Proposition 3.2, a small $\mathrm{UC}(f,\mathcal{U}_{\mathrm{TCE}})$ guarantees that threshold-based decisions on top-class confidence are robust to monotonic recalibration, and by Proposition 3.3 that this confidence is a calibrated predictor of actual top-class accuracy; analogous guarantees hold for $\mathrm{UC}(f,\mathcal{U}_{\mathrm{CWE}})$.

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

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

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

Alternatively, in applications like information retrieval or recommender systems (Järvelin and Kekäläinen, 2002), the realized utility depends on the rank assigned to the true outcome $Y=e_j$. Given a model's prediction $p=f(X)$, assuming $p_1,\dots,p_C$ are distinct (or that ties are broken arbitrarily/randomly among equal coordinates), the rank of class j, denoted $\mathrm{rank}(p,j)$, is its position across p, i.e. $\mathrm{rank}(p,j)\coloneqq\sum_{i\in[C]}\mathbf{1}\{p_j\le p_i\}$. Using a valuation vector $\theta\in[-1,1]^C$, a rank-based utility function can then be constructed as $u_\theta(p,e_j)=\theta_{\mathrm{rank}(p,j)}$ with the associated expected utility function $v_{u_\theta}(X)=\sum_{i=1}^C f(X)_i\theta_{\mathrm{rank}(f(X),i)}$. Calibrating for such utilities ensures the model's expected rank-based performance aligns with reality. A prominent special case is topK utility, where the valuation vector $\theta^{(K)}$ for a given $K\in[C]$ is defined such that $\theta^{(K)}_r=1$ if $r\le K$ and $\theta^{(K)}_r=0$ if $r>K$.

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

As discussed in Section 2, decision calibration (Zhao et al., 2021) ensures that for problems with up to K actions, the model's predicted utility for its recommended action matches the actual realized utility. We can frame a similar guarantee within utility calibration. For any bounded loss function $l : \mathcal{Y} \times [K] \to [-1, 1]$ 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 function is then $u_l(p, y) = -l(y, \delta_l(p))$, representing the negative loss from outcome y under action $\delta_l(p)$. The predicted expected utility is $v_{u_l}(X) = -\mathbb{E}_{\hat{Y} \sim f(X)}[l(\hat{Y}, \delta_l(f(X)))|X]$.

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

These aforementioned examples illustrate that calibrating against classes $\mathcal{U}$ provides guarantees tailored to diverse user needs, moving beyond simplistic binarized assessments.

4 SCALABLE EVALUATION OF UTILITY CALIBRATION

Estimating $\sup_{u \in \mathcal{U}} \text{UC}(f, u)$ in (3.5) presents two key challenges: the *computational complexity* of the optimization, and the *sample complexity* required for the empirical supremum to converge to its true value. We introduce the two notions of proactive and interactive measurability to decouple these two aspects.

Definition 4.1 (Proactive Measurability)**.** *The utility calibration error w.r.t. class* $\mathcal{U}$ *is proactively measurable if there exists an algorithm A and polynomial functions* N_{poly}, T_{poly} *such that for any* $\varepsilon, \delta > 0$ *and* $n \ge N_{poly}(C, 1/\varepsilon, 1/\delta)$ *samples* $S \sim D^n$*, algorithm* $A(S)$ *outputs* $\hat{u}$ *satisfying* $|\text{UC}(f, \hat{u}) - \text{UC}(f, \mathcal{U})| \le \varepsilon$ *with probability at least* $1 - \delta$ *and the runtime of* $A(S)$ *is bounded by* $T_{poly}(C, n)$*.*

Generally, for a finite class $\mathcal{U}$, if $|\mathcal{U}|$ grows polynomially in C then by Lemma 3.4 we can guarantee proactive measurability. Nonetheless, even for simple infinite classes such as $\mathcal{U}_{\text{lin}}$, proactive measurability reduces to a non-convex optimization problem that cannot be generally solved in polynomial time. In fact, even aiming for a weaker notion, namely *improper auditing*, Gopalan et al. (2024) showed that assessing both weaker and stronger notions than $\text{UC}(f, \mathcal{U}_{\text{lin}})$ cannot be done in polynomial time in both the error ε^{-1} and the dimension C (Gopalan et al., 2024, Theorem 1.3, Theorem 5.2, and Theorem 8.6). A more detailed description of Gopalan et al. (2024) hardness results is in Appendix B.3. Next, we thus propose an alternative criterion of measurability that decouples the statistical guarantee from the computational complexity of verifying the supremum.

Definition 4.2 (Interactive Measurability)**.** *The utility calibration error w.r.t. class* $\mathcal{U}$ *is interactively measurable if there exists an estimator* $\widehat{\text{UC}}(f, u; S)$ *and a polynomial function* N_{poly} *such that for* $n \ge N_{poly}(C, 1/\varepsilon, 1/\delta)$ *samples* $S \sim D^n$*, it holds with probability at least* $1 - \delta$ *that* $\sup_{u \in \mathcal{U}} |\widehat{\text{UC}}(f, u; S) - \text{UC}(f, u)| \le \varepsilon$*.*

Interactive measurability represents a much more achievable goal. For example, while decision calibration is computationally hard to measure, Zhao et al. (2021) showed that it admits polynomial sample complexity. We establish the following result for the key utility classes of interest. We defer the proof to Appendix B.4.

Corollary 4.3 (Interactive Measurability of $\mathcal{U}_{\text{lin}}$ and $\mathcal{U}_{\text{rank}}$)**.** *The utility calibration error is interactively measurable for both* $\mathcal{U}_{\text{lin}}$ *and* $\mathcal{U}_{\text{rank}}$*. Concretely, with* $n = \tilde{O}\big((C + \log(1/\delta))/\varepsilon^2\big)$ *i.i.d. samples, it holds with probability at least* $1 - \delta$ *that*

$$\sup_{u \in \mathcal{U}_{\text{lin}}} \left|\widehat{\text{UC}}(f, u; S) - \text{UC}(f, u)\right| \le \varepsilon.$$

More generally, a utility class $\mathcal{U}$ is interactively measurable whenever the functions $X \mapsto u(f(X), e_i)\mathbf{1}\{v_u(X) \in I\}$ for $u \in \mathcal{U}$, $i \in [C]$, $I \in \mathbb{I}[-1, 1]$ admit controlled Rademacher complexity; see Appendix B.4 for the general theorem.

In summary, while proactively measuring the worst-case utility calibration error $\text{UC}(f, \mathcal{U})$ is computationally prohibitive for expressive utility classes, interactive measurability allows efficient estimation of $\text{UC}(f, u)$ uniformly for any specific $u \in \mathcal{U}$. We leverage this distinction to propose a scalable evaluation methodology that aims to characterize the *distribution* of utility calibration errors across $\mathcal{U}$, providing a more nuanced picture of reliability over a spectrum of utilities.

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 $\widehat{E}_{m,n} \coloneqq \widehat{\mathrm{UC}}(f, u_m; S)$ using n data points from a sample S. These M error estimates then form an *empirical Cumulative Distribution Function (eCDF)*, $\widehat{F}_{E,M,n}(e) \coloneqq \frac{1}{M}\sum_{m=1}^{M} \mathbf{1}\{\widehat{E}_{m,n} \le e\}$, which serves as an empirical proxy for the true CDF, $F_E(e) \coloneqq \mathbb{P}_{u\sim\mathcal{D}_{\mathcal{U}}}(\mathrm{UC}(f,u) \le e)$. We provide guarantees on the difference between $F_E(e)$ and $\hat{F}_{E,M,n}(e)$ in Appendix B.5. Informally, if individual utility-calibration errors can be uniformly estimated to accuracy $\varepsilon_{\mathrm{stat}}$ from n samples, then with high probability over the draw of the dataset and the M sampled utilities, the deviation between the true CDF F_E and the empirical eCDF $\widehat{F}_{E,M,n}$ in L_2 scales as

$$\tilde{O}\left(\sqrt{\varepsilon_{\mathrm{stat}}} + \sqrt{\tfrac{1}{M}}\right).$$

In particular, $\mathcal{U}_{\mathrm{lin}}$ (Example 3.6) and $\mathcal{U}_{\mathrm{rank}}$ (Example 3.7) both admit finite-dimension parameterization. For $\mathcal{U}_{\mathrm{lin}}$, we construct $\mathcal{D}_{\mathcal{U}_{\mathrm{lin}}}$ by sampling the payoff vectors a uniformly in $\partial B_\infty \coloneqq \{a \in \mathbb{R}^C : \|a\|_\infty = 1\}$. Meanwhile, for $\mathcal{U}_{\mathrm{rank}}$, we also sample from $\mathcal{D}_{\mathcal{U}_{\mathrm{rank}}}$ by uniformly sampling valuation vectors ∂B_∞, which satisfy $\theta_1 \ge \theta_2 \ge \cdots \ge \theta_C$. This is to reflect a rational preference for better ranks, i.e. the higher the rank of the true realization within the predictions of $f(X)$, the higher the utility.

4.1 Post-Hoc Calibration via Utility-Aware Patching

Since $\mathrm{UC}(f,u)$ is a form of weighted calibration (Equation (3.3)), standard patching algorithms from the (multi)calibration literature (Hébert-Johnson et al., 2018; Gopalan et al., 2022a; Jung et al., 2021; Duchi, 2024) apply directly. Each iteration of Algorithm 1 proceeds in two steps. First, a *witness* $w_t \in \mathcal{W}(\mathcal{U})$ is identified as the worst-case direction of miscalibration: it corresponds to the utility $u \in \mathcal{U}$ and interval I for which the expected inner product between the prediction error $f^{(t)}(X) - Y$ and the witness weight $w_t(f^{(t)}(X))$ is largest. This is the direction in which the current predictor most systematically over- or underestimates the utility. Second, the predictor is updated by a gradient step along this direction and projected back onto the simplex Δ^{C-1}.

The algorithm terminates as soon as no witness can certify a miscalibration error above ε, at which point $\mathrm{UC}(f^{(t)},\mathcal{U}) \le \varepsilon$ by definition. The Brier score serves as a Lyapunov function that strictly decreases at each step with $\mathrm{err}_t > \varepsilon$, bounding the total number of iterations.

Proposition 4.4 (Convergence and Brier Score Guarantee; Hébert-Johnson et al., 2018; Gopalan et al., 2024)**.** *With stepsize $\eta_t = \mathrm{err}_t/C$, Algorithm 1 terminates in $T = O(C/\varepsilon^2)$ iterations, with the Brier score $\mathbb{E}[\|Y - f^{(t)}(X)\|_2^2]$ non-increasing at every step.*

Algorithm 1 Iterative Patching for Utility Calibration

Input: Predictor $f^{(0)}$, witness class $\mathcal{W}(\mathcal{U})$, tolerance $\varepsilon > 0$. **Set** $t \leftarrow 0$.
loop
  $w_t \in \mathrm{argmax}_{w\in\mathcal{W}(\mathcal{U})} \mathbb{E}[\langle f^{(t)}(X) - Y,\, w(f^{(t)}(X))\rangle]$.
  $\mathrm{err}_t \leftarrow \mathbb{E}[\langle f^{(t)}(X) - Y, w_t(f^{(t)}(X))\rangle]$.
  if $\mathrm{err}_t \le \varepsilon$ **then**
    break
  end if
  $f^{(t+1)}(X) \leftarrow \pi_{\Delta^{C-1}}\big(f^{(t)}(X) - \eta_t w_t(f^{(t)}(X))\big)$.
  $t \leftarrow t + 1$.
end loop
Return $f^{(t)}$.

Method	Brier Score ($\times 10^2$)	$\mathrm{CWE}^{\mathrm{bin}}$ ($\times 10^4$)	$\mathrm{TCE}^{\mathrm{bin}}$ ($\times 10^3$)	$\mathcal{U}_{\mathrm{comb}}$ ($\times 10^3$)
Uncalibrated	22.6 ± 0.29	2.46 ± 0.0212	94.2 ± 2.03	124.0 ± 0.924
Dirichlet	$\mathbf{21.3 \pm 0.228}$	1.34 ± 0.0392	13.7 ± 1.37	26.1 ± 0.67
IR	22.9 ± 0.235	$\mathbf{1.1 \pm 0.0164}$	33.1 ± 1.59	54.1 ± 0.94
Temp. Scaling	21.8 ± 0.252	1.26 ± 0.0214	30.0 ± 2.55	45.2 ± 0.53
Vector Scaling	22.8 ± 0.238	1.54 ± 0.0395	35.1 ± 2.53	37.4 ± 2.13
Patching	21.6 ± 0.31	1.56 ± 0.0432	$\mathbf{10.3 \pm 2.04}$	$\mathbf{19.4 \pm 2.55}$

Table 1: `ViT`-ImageNet-1K results. Mean ± 2 std. errors over 10 splits.

5 EXPERIMENTS

We evaluate on ImageNet-1K (Deng et al., 2009) with a pretrained Vision Transformer (`ViT`) (Dosovitskiy et al., 2021), comparing Temperature Scaling (Platt et al., 1999), Vector Scaling (Kull et al., 2017), Dirichlet recalibration (Kull et al., 2019), and Isotonic Regression (Zadrozny and Elkan, 2002). As our post-hoc baseline, we apply Algorithm 1 with the combined class $\mathcal{U}_{\mathrm{comb}} \coloneqq \mathcal{U}_{\mathrm{CWE}} \cup \mathcal{U}_{\mathrm{topK}}$. Appendix C provides extended results: additional architectures, datasets, and modalities, sensitivity to aligned vs. misaligned utilities, additional utility families including discounted cumulative gain (Järvelin and Kekäläinen, 2002) and hierarchical classification loss (Deng et al., 2012), and robustness of eCDF conclusions to the sampling prior.

In Table 1, we compare Brier score, binned binarized metrics $\mathrm{TCE}_{\mathrm{binned}}$ and $\mathrm{CWE}_{\mathrm{binned}}$ (with 15 equal-weight bins), and a combined utility calibration metric w.r.t. $\mathcal{U}_{\mathrm{comb}}$. Since both $\mathcal{U}_{\mathrm{CWE}}$ and $\mathcal{U}_{\mathrm{topK}}$ are finite classes, $\mathcal{U}_{\mathrm{comb}}$ is measured exactly without sampling. As expected, all post-hoc methods improve Brier and calibration errors. No single method dominates uniformly; the patching algorithm achieves the best top-

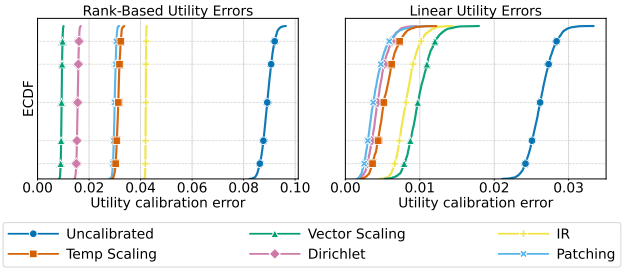

Figure 1: eCDF of utility calibration errors for `ViT` on ImageNet-1K ($\mathcal{U}_{\text{rank}}$ and $\mathcal{U}_{\text{lin}}$, M=1500 utilities). Curves further left indicate better calibration.

class binned calibration error and $\mathcal{U}_{\text{comb}}$.

Figure 1 displays eCDFs for broader utility classes: rank-based ($\mathcal{U}_{\text{rank}}$) and linear ($\mathcal{U}_{\text{lin}}$), sampled as described in Section 4. The uncalibrated model is consistently worst; post-hoc methods shift the curves left to varying degrees. Crucially, the eCDF surfaces trends that aggregate scores in Table 1 can obscure: how tightly errors concentrate around the median, whether heavy tails persist, and whether gains are uniform across the utility class. Performance is also not uniform across utility families: Vector Scaling is best on $\mathcal{U}_{\text{rank}}$ but worst on $\mathcal{U}_{\text{lin}}$, underscoring the importance of matching the calibrator to the target utility class.

6 CONCLUSION

Utility calibration provides a unified, application-centric framework for evaluating classifier reliability. Its specific instantiations $\mathcal{U}_{\text{TCE}}$ and $\mathcal{U}_{\text{CWE}}$ are binning-free alternatives to traditional metrics with concrete decision-theoretic guarantees, and the eCDF evaluation across broader utility classes surfaces distributional insights that single-metric summaries obscure. Together, the scalable assessment methodology and the patching algorithm give practitioners a modular toolkit that can be adapted to domain-specific utility classes.

Limitations. The framework requires the user to specify or sample from a utility class $\mathcal{U}$; when the true downstream utility is entirely unknown, the choice of $\mathcal{U}$ is a modelling decision that may not reflect all failure modes. Interactive measurability ensures uniform estimation, but finding the *worst-case* $u \in \mathcal{U}$ proactively remains computationally intractable for expressive classes. In particular, practitioners must either restrict $\mathcal{U}$ or rely on the eCDF summary. Finally, while the patching algorithm provably reduces the utility calibration error, it assumes i.i.d. data and an oracle for the calibration set; distribution shift between calibration and deployment can limit its effectiveness.

Acknowledgments

The work of Aymeric Dieuleveut and Mahmoud Hegazy is supported by French State aid managed by the Agence Nationale de la Recherche (ANR) under the France 2030 program with the reference ANR-23-PEIA-005 (REDEEM project), and ANR-23-IACL-0005, in particular the Hi!Paris FLAG chair. Additionally, this project was funded by the European Union (ERC-2022-SYG-OCEAN-101071601). Views and opinions expressed are, however, those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them. This publication is part of the Chair "Markets and Learning," supported by Air Liquide, BNP PARIBAS ASSET MANAGEMENT Europe, EDF, Orange and SNCF, sponsors of the Inria Foundation.

References

Juozas Vaicenavicius, David Widmann, Carl Andersson, Fredrik Lindsten, Jacob Roll, and Thomas Schön. Evaluating model calibration in classification. In *The 22nd international conference on artificial intelligence and statistics*, pages 3459–3467. PMLR, 2019.

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.

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.

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.

Alexandre B Tsybakov. Nonparametric estimators. *Introduction to Nonparametric Estimation*, pages 1–76, 2009.

Chirag Gupta and Aaditya Ramdas. Top-label calibration and multiclass-to-binary reductions. In *International Conference on Learning Representations*, 2022.

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.

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.

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.

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.

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.

Parikshit Gopalan, Lunjia Hu, and Guy N Rothblum. On computationally efficient multi-class calibration. *arXiv preprint arXiv:2402.07821*, 2024.

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.

David Widmann, Fredrik Lindsten, and Dave Zachariah. Calibration tests in multi-class classification: A unifying framework. *Advances in neural information processing systems*, 32, 2019.

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.

Charles Elkan. The foundations of cost-sensitive learning. In *International joint conference on artificial intelligence*, volume 17, pages 973–978. Lawrence Erlbaum Associates Ltd, 2001.

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.

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.

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.

Bianca Zadrozny and Charles Elkan. Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. In *Icml*, volume 1, pages 609–616, 2001.

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.

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.

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.

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.

Charlie Marx, Sofian Zalouk, and Stefano Ermon. Calibration by distribution matching: Trainable kernel calibration metrics. *Advances in Neural Information Processing Systems*, 36, 2024.

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.

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.

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.

Jeremy Nixon, Michael W Dusenberry, Linchuan Zhang, Ghassen Jerfel, and Dustin Tran. Measuring calibration in deep learning.

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.

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.

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_jIy5QFB7.

Parikshit Gopalan, Michael P Kim, Mihir A Singhal, and Shengjia Zhao. Low-degree multicalibration. In *Conference on Learning Theory*, pages 3193–3234. PMLR, 2022a.

Georgy Noarov, Ramya Ramalingam, Aaron Roth, and Stephan Xie. High-dimensional prediction for sequential decision making. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=uRAgIVnAO6.

Aaron Roth and Mirah Shi. Forecasting for swap regret for all downstream agents. In *Proceedings of the 25th ACM Conference on Economics and Computation*, pages 466–488, 2024.

Bobby Kleinberg, Renato Paes Leme, Jon Schneider, and Yifeng Teng. U-calibration: Forecasting for an unknown agent. In *The Thirty Sixth Annual Conference on Learning Theory*, pages 5143–5145. PMLR, 2023.

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.

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, 2022b.

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, 2023a.

Parikshit Gopalan, Michael Kim, and Omer Reingold. Swap agnostic learning, or characterizing omniprediction via multicalibration. *Advances in Neural Information Processing Systems*, 36:39936–39956, 2023b.

Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446, 2002.

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.

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.

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.

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.

Jia Deng, Jonathan Krause, Alexander C Berg, and Li Fei-Fei. Hedging your bets: Optimizing accuracy-specificity trade-offs in large scale visual recognition. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3450–3457. IEEE, 2012.

Peter L Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of machine learning research*, 3(Nov):463–482, 2002.

VN Vapnik and A Ya Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability & Its Applications*, 16(2):264–280, 1971.

Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. Foundations of machine learning. adaptive computation and machine learning, 2018.

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.

Andrey Kupavskii. The vc-dimension of k-vertex d-polytopes. *Combinatorica*, 40(6):869–874, 2020.

Martin Anthony and Peter L Bartlett. *Neural network learning: Theoretical foundations*. cambridge university press, 2009.

Paul Goldberg and Mark Jerrum. Bounding the vapnik-chervonenkis dimension of concept classes parameterized by real numbers. In *Proceedings of the sixth annual conference on Computational learning theory*, pages 361–369, 1993.

Pascal Massart. The tight constant in the dvoretzky-kiefer-wolfowitz inequality. *The annals of Probability*, pages 1269–1283, 1990.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images, 2009.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28, 2015.

Yaofo Chen. pytorch-cifar-models: Pretrained models for CIFAR10/100 in PyTorch, February 2020. URL https://github.com/chenyaofo/pytorch-cifar-models.

Ross Wightman. PyTorch Image Models. URL https://github.com/huggingface/pytorch-image-models.

Tiago Salvador. Calibration baselines. https://github.com/tiagosalvador/calibration-baselines, 8 2022. URL https://github.com/tiagosalvador/calibration-baselines. Last commit August 2022. Accessed: May 22, 2025.

Tim Head, MechCoder, Gilles Louppe, Iaroslav Shcherbatyi, fcharras, Zé Vinícius, cmmalone, Christopher Schröder, nel215, Nuno Campos, Todd Young, Stefano Cereda, Thomas Fan, rene rex, Kejia (KJ) Shi, Justus Schwabedal, carlosdanielcsantos, Hvass-Labs, Mikhail Pak, SoManyUsernamesTaken, Fred Callaway, Loïc Estève, Lilian Besson, Mehdi Cherti, Karlson Pfannschmidt, Fabian Linzberger, Christophe Cauet, Anna Gut, Andreas Mueller, and Alexander Fabisch. scikit-optimize/scikit-optimize: v0.5.2, March 2018. URL https://doi.org/10.5281/zenodo.1207017.

Mathieu Blondel, Quentin Berthet, Marco Cuturi, Roy Frostig, Stephan Hoyer, Felipe Llinares-López, Fabian Pedregosa, and Jean-Philippe Vert. Efficient and modular implicit differentiation. *arXiv preprint arXiv:2105.15183*, 2021.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

Radim Rehurek and Petr Sojka. Gensim–python framework for vector space modelling. *NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic*, 3(2), 2011.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes/No/Not Applicable] **Answer: Yes.** Sections 3 and 4 define the mathematical setting, utility classes, and evaluation criteria; Algorithm 1 gives the post-hoc calibration procedure.
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes/No/Not Applicable] **Answer: Yes.** Lemma 3.4, Corollary 4.3, Proposition 4.4, and Appendix A provide computational, sample-complexity, and convergence analyses.
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes/No/Not Applicable] **Answer: Yes.** The source code is included in the supplementary material, and Appendix C.1 specifies the main software dependencies used for the experiments.

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes/No/Not Applicable] **Answer: Yes.** The assumptions are stated with each result, including bounded utilities, i.i.d. sampling, and continuity conditions when needed.
   (b) Complete proofs of all theoretical results. [Yes/No/Not Applicable] **Answer: Yes.** Complete proofs are provided in Appendix A and Appendix B.
   (c) Clear explanations of any assumptions. [Yes/No/Not Applicable] **Answer: Yes.** The assumptions are discussed around the relevant results in Sections 3–4 and in the appendix proofs.

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes/No/Not Applicable] **Answer: Yes.** The code is included in the supplementary material; Appendix C.1 describes the datasets, pretrained checkpoints, calibration/test split protocol, and calibration methods.

(b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes/No/Not Applicable] **Answer: Yes.** Appendix C.1 gives the 70/30 calibration/test split, the 10 repeated splits, the post-hoc calibration procedures, and the hyperparameter-tuning protocol.

(c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes/No/Not Applicable] **Answer: Yes.** Section 5 and Appendix C.1 define the reported metrics; Table 1 reports mean $\pm$ two standard errors over 10 splits, and figure captions specify the utility-sampling setup for eCDF plots.

(d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes/No/Not Applicable] **Answer: Yes.** Appendix C.1 reports that all experiments, including hyperparameter tuning, took approximately 120 hours on an A100 80GB NVIDIA GPU.

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

(a) Citations of the creator If your work uses existing assets. [Yes/No/Not Applicable] **Answer: Yes.** Section 5 and Appendix C.1 cite the datasets, pretrained models, calibration baselines, and software libraries used in the experiments.

(b) The license information of the assets, if applicable. [Yes/No/Not Applicable] **Answer: Yes.** Appendix C.1 lists the licenses or usage terms for the datasets and publicly available models where applicable.

(c) New assets either in the supplemental material or as a URL, if applicable. [Yes/No/Not Applicable] **Answer: Yes.** The experimental source code is included in the supplementary material.

(d) Information about consent from data providers/curators. [Yes/No/Not Applicable] **Answer: Not Applicable.** The experiments use established public benchmark datasets and pretrained checkpoints rather than newly collected human-subject data.

(e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Yes/No/Not Applicable] **Answer: Not Applicable.** The work does not curate or release new sensitive-content datasets and uses existing classification benchmarks only for aggregate calibration evaluation.

5. If you used crowdsourcing or conducted research with human subjects, check if you include:

(a) The full text of instructions given to participants and screenshots. [Yes/No/Not Applicable] **Answer: Not Applicable.**

(b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Yes/No/Not Applicable] **Answer: Not Applicable.**

(c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Yes/No/Not Applicable] **Answer: Not Applicable.**

Scalable Utility-Aware Multiclass Calibration: Supplementary Materials

Organization of the Appendix

The appendix is organized as follows: Appendix A contains the proof of Proposition 4.4 and the empirical implementation details for the patching algorithm; Appendix B collects deferred material, including CutOff calibration (Appendix B.1), bounds for binned estimators (Appendix B.2), computational aspects of proactive measurability (Appendix B.3), sample complexity bounds for interactive measurability (Appendix B.4), and eCDF guarantees (Appendix B.5). The subsequent subsections present proofs of statements that were explicitly stated in the main body. Finally, Appendix C provides additional experiments, including robustness of eCDF conclusions to the sampling prior (Appendix C.3).

Appendix Contents

A Proof of Proposition 4.4 and Empirical Implementation

Algorithm 1 and Proposition 4.4 in Section 4.1 present the patching algorithm and its convergence guarantee. We prove Proposition 4.4 here and then discuss the empirical (finite-sample) implementation.

Proof of Proposition 4.4. We include the proof for completeness. It follows the approach of Hébert-Johnson et al. (2018) 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. (2024). A more general version can be found in Duchi (2024, Chapter 15). The change in Brier score $L(f) = \mathbb{E}[\|Y - f(X)\|_2^2]$ from $f^{(t)}$ to $f^{(t+1)}$ is:

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

Each witness $w \in \mathcal{W}(\mathcal{U})$ is of the form $p \mapsto \xi \vec{u}(p) \mathbf{1}\{v_u(p) \in I\}$ 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 $\eta_t = \mathrm{err}_t / C$:

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

This proves that the Brier score does not increase and strictly decreases if $\mathrm{err}_t > 0$. If the algorithm continues, it is because $\mathrm{err}_t > \varepsilon$, 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 at step $\mathrm{err}_t > \varepsilon$ decreases $L(f)$ by at least ε^2/C, the algorithm must terminate in at most $O\left(C/\varepsilon^2\right)$ such steps. □

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 4.4 (i.e., termination in $O(C/\varepsilon^2)$ iterations) can be extended to this empirical setting, provided that the empirically estimated error $\widehat{\mathrm{err}}_t := \hat{\mathbb{E}}_N[\langle f^{(t)}(X) - Y, w_t(f^{(t)}(X))\rangle]$ is a sufficiently accurate approximation of the true error err_t. Specifically, if $\widehat{\mathrm{err}}_t$ is within $O(\varepsilon)$ of err_t whenever $\mathrm{err}_t > \varepsilon$, 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 $\mathcal{U}_{\mathrm{TCE}}$, where $|\mathcal{U}| = 1$, using Lemma 3.4, $N \leq \tilde{O}\left(\frac{\log(1/\delta)}{\varepsilon^2}\right)$. Similarly, for Class-Wise utility $\mathcal{U}_{\mathrm{CWE}}$ and Top-K utility $\mathcal{U}_{\mathrm{topK}}$, $|\mathcal{U}| = C$ in both cases; 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 ($Y \in \{0, 1\}, f : \mathcal{X} \to [0, 1]$), Rossellini et al. (2025) demonstrate that if the metric

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

is small, then a simple decision rule $\widehat{Y}_\tau : X \to \{0,1\}$ of the form $\hat{Y}_\tau = \mathbf{1}\{f(X) \geq \tau\}$ evaluated against its associated binary decision loss, cannot be substantially improved by monotonic post-hoc calibration. More concretely, let $R_{\mathrm{bd}}(g;\tau) := \mathbb{E}[\ell_{\mathrm{bd}}(Y, \hat{Y}_\tau; \tau)]$ be the risk under the binary decision loss $\ell_{\mathrm{bd}}(Y, \widehat{Y}; \tau) = \tau(1-Y)\widehat{Y} + (1-\tau)Y(1-\widehat{Y})$. Then, Rossellini et al. (2025, Prop. 3.2) show that for any $\tau \in [0,1]$:

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

This guarantee implies that if $\Delta_{\mathrm{Cutoff}}(f)$ is small, the decision-maker, thresholding $f(X)$ to make a binary decision, gains little by applying any monotonic recalibration to $f(X)$. Similarly, they showed that in the binary case, CutOff calibration error can be used to bound distance from calibration. As such, Propositions 3.2 and 3.3 extend the results of Rossellini et al. (2025) to the multiclass setting.

B.2 Bounding Binned Estimators using Utility Calibration

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

$$\mathrm{TCE}^{\mathrm{bin}}(f) = \sum_{j=1}^{m} \left|\mathbb{E}\left[(p_X - Y_X)\,\mathbf{1}\{p_X \in B_j\}\right]\right|,$$
$$\mathrm{UC}(f, \mathcal{U}_{\mathrm{TCE}}) = \sup_{I\in\mathbb{I}[0,1]} \left|\mathbb{E}\left[(Y_X - p_X)\,\mathbf{1}\{p_X \in I\}\right]\right|.$$

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

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 $\mathbb{E}[Y_X | p_X = 0.55] = 0.95$. If $\mathrm{TCE}^{\mathrm{bin}}(f)$ uses bins $B_1 = [0, 1/3), B_2 = [1/3, 2/3), B_3 = [2/3, 1]$, the expected error is 0. Thus, $\mathrm{TCE}^{\mathrm{bin}}(f) = 0$.

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

B.3 Hardness of Proactive Measurability

Proactive measurability for a utility class $\mathcal{U}$, as defined in Definition 4.1, necessitates an algorithm to efficiently find $\hat{u} \in \mathcal{U}$, whose utility calibration error $\mathrm{UC}(f, \hat{u})$ approximates $\sup_{u\in\mathcal{U}} \mathrm{UC}(f, u)$. This is equivalent to efficiently finding an approximate worst-case function from the witness class $\mathcal{W}(\mathcal{U}) = \bigcup_{u\in\mathcal{U}} \{X \mapsto \xi\vec{u}(X)\mathbf{1}\{v_u(X) \in I\} \mid I \in \mathbb{I}[-1,1], \xi \in \{-1,1\}\}$, given that the worst-case interval for a fixed u is efficiently findable (Lemma 3.4). The work of Gopalan et al. (2024) establishes computational hardness for "auditing with a witness" for related, expressive classes of witness functions.

Definition B.1 (Auditing with a witness (Gopalan et al., 2024))**.** *An (α, β) auditor for a witness class $\mathcal{W}_{target}$ is an algorithm that, when given access to a distribution $\mathcal{D}$ where $\mathrm{CE}_{\mathcal{W}_{target}}(\mathcal{D}) > \alpha$, returns any function $w' : \Delta^{C-1} \to [-1,1]^C$ such that $\mathbb{E}_{(\mathcal{X},\mathcal{Y})\sim\mathcal{D}}[\langle Y - f(X), w'(f(X))\rangle] \geq \beta$.*

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

Gopalan et al. (2024) demonstrate hardness for two key notions.

1. First, for decision calibration, their witness class $\mathcal{W}_{\mathrm{dec}}$ involves partitioning Δ^{C-1} using hyperplanes, i.e. $\mathcal{W}_{\mathrm{dec}} \coloneqq \{x \to g'\mathbf{1}\{a^T x \geq b\} + g\mathbf{1}\{a^T x < b\} \mid g', g, a \in \mathbb{R}^C, b \in \mathbb{R}, \text{ s.t. } \|g'\|_2, \|g\|_2 \leq 1\}$. Auditing for $\mathcal{W}_{\mathrm{dec}}$ is shown to be computationally hard under standard assumptions (Gopalan et al. (2024, Thm. 5.1)), i.e. cannot be performed in polynomial time for non-trivial α under standard computational complexity assumptions. Up to a scaling, this is a slightly more general class than $\mathcal{U}_{\mathrm{lin}}$.

2. Second, Gopalan et al. (2024) introduced projected smooth calibration, which is very similar to our notion of utility calibration for $\mathcal{U}_{\text{lin}}$, but replaced the hard interval indicator with Lipschitz functions. Again, auditing for this notion was also proven computationally hard, in the sense that no auditing algorithm can be polynomial in $1/\alpha$ (Gopalan et al., 2024, Thm. 8.1).

B.4 Interactive Measurability

In this section, we establish conditions under which the utility calibration error $\mathrm{UC}(f, \mathcal{U})$ is interactively measurable, completing Section 4. This involves bounding the sample complexity required for the empirical estimates $\widehat{\mathrm{UC}}(f, u; S)$ to uniformly converge to their true values $\mathrm{UC}(f, u)$ over all $u \in \mathcal{U}$. Throughout this section, we assume that $\hat{\mathbb{E}}_n$ denotes the empirical expectation over n i.i.d. samples.

First, we start by bounding the Rademacher complexity of the class of functions

$$\mathcal{G}_{\mathcal{U}} \coloneqq \left\{ (X, Y) \mapsto \langle Y - f(X), \vec{u}(X) \mathbf{1}\left\{v_u(X) \in I\right\} \rangle \;\middle|\; u \in \mathcal{U},\ I \in \mathbb{I}\left[-1, 1\right] \right\} \tag{B.2}$$

with respect to its coordinate-wise components $\mathcal{P}_1, \ldots, \mathcal{P}_C$, where

$$\mathcal{P}_j(\mathcal{U}) \coloneqq \left\{ X \mapsto u(f(X), e_j) \mathbf{1}\left\{v_u(X) \in I\right\} \mid u \in \mathcal{U}, I \in \mathbb{I}\left[-1, 1\right] \right\}. \tag{B.3}$$

This generic bound may be transformed into a sample complexity bound for interactive measurability by Corollary B.5. As an example, we apply it to the class of linear utility functions $\mathcal{U}_{\text{lin}}$. Nonetheless, in Appendix B.4.1, we derive tighter bounds for $\mathcal{U}_{\text{lin}}$ and $\mathcal{U}_{\text{rank}}$ by directly bounding their pseudo-dimension. Before proceeding, we first recall standard definitions and results.

Definition B.2. *(Bartlett and Mendelson, 2002, Rademacher Complexity) Let $\mathcal{F}$ be a class of real-valued functions $h : \mathcal{Z} \to \mathbb{R}$. Given n samples $S_Z = (Z_1, \ldots, Z_n)$ where $Z_i \sim D_Z$, the empirical Rademacher complexity of $\mathcal{F}$ given S_Z is*

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

where σ_i are i.i.d. Rademacher random variables. In addition, the expected Rademacher complexity is defined as

$$\mathfrak{R}_n(\mathcal{F}) = \mathbb{E}_{S_Z}[\hat{\mathfrak{R}}_n(\mathcal{F}|S_Z)].$$

Definition B.3. *(Vapnik and Chervonenkis, 1971, VC dimension) Let $\mathcal{H}$ be a class of binary-valued functions on a domain $\mathcal{Z}$. We say that $\mathcal{H}$ shatters a set $\{z_1, \ldots, z_m\} \subseteq \mathcal{Z}$ if for every labeling $b \in \{0, 1\}^m$ there exists $h \in \mathcal{H}$ such that $h(z_i) = b_i$ for all i. The VC dimension $\mathrm{VC}(\mathcal{H})$ is the largest m for which some set of size m is shattered (or ∞ if no such largest m exists).*

Finally, we list a common result combining Massart's lemma, which bounds the Rademacher complexity using the growth function (Mohri et al., 2018, Theorem 3.7) and Sauer's lemma, which bounds the growth function (Mohri et al., 2018, Theorem 3.17).

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

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

Theorem B.4 (Rademacher Complexity Bound for $\mathcal{G}_{\mathcal{U}}$)**.** *For a utility class $\mathcal{U}$, the Rademacher complexity of $\mathcal{G}_{\mathcal{U}}$ is bounded as:*

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

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

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

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

$$\vec{\mathcal{W}}_{\mathcal{U}} \coloneqq \left\{ X \mapsto \vec{u}(X)\mathbf{1}\left\{v_u(X) \in I\right\} \;\middle|\; u \in \mathcal{U},\ I \in \mathbb{I}[-1,1] \right\}. \tag{B.4}$$

The functions $w \in \vec{\mathcal{W}}_{\mathcal{U}}$ map to $[-1,1]^C$. Using Maurer (2016, Corollary 4), we have

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

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 $S_X = \{X_k\}_{k=1}^n$. It is defined as:

$$\hat{\mathfrak{R}}_n^{\text{vec}}(\vec{\mathcal{W}}_{\mathcal{U}}|S_X) \coloneqq \mathbb{E}_{\boldsymbol{\sigma}}\left[\sup_{w\in\vec{\mathcal{W}}_{\mathcal{U}}} \frac{1}{n}\sum_{k=1}^n \langle \boldsymbol{\sigma}_k, w(X_k)\rangle\right]. \tag{B.5}$$

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

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

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

$$\begin{aligned}
\hat{\mathfrak{R}}_n^{\text{vec}}(\vec{\mathcal{W}}_{\mathcal{U}}|S_X) &= \mathbb{E}_{\boldsymbol{\sigma}}\left[\sup_{\substack{u\in\mathcal{U}\\ I\in\mathbb{I}[-1,1]}} \frac{1}{n}\sum_{k=1}^n\sum_{j=1}^C \sigma_{kj}u(f(X_k),e_j)\mathbf{1}\left\{v_u(X_k)\in I\right\}\right] \quad \text{by eq.(B.4) and (B.5)} \\
&\le \mathbb{E}_{\boldsymbol{\sigma}}\left[\sum_{j=1}^C \sup_{\substack{u\in\mathcal{U}\\ I\in\mathbb{I}[-1,1]}} \frac{1}{n}\sum_{k=1}^n \sigma_{kj}u(f(X_k),e_j)\mathbf{1}\left\{v_u(X_k)\in I\right\}\right] \quad \text{(subadditivity of sup)} \\
&= \sum_{j=1}^C \mathbb{E}_{\boldsymbol{\sigma}_{\cdot j}}\left[\sup_{\substack{u\in\mathcal{U}\\ I\in\mathbb{I}[-1,1]}} \frac{1}{n}\sum_{k=1}^n \sigma_{kj}u(f(X_k),e_j)\mathbf{1}\left\{v_u(X_k)\in I\right\}\right] \\
&= \sum_{j=1}^C \hat{\mathfrak{R}}_n(\mathcal{P}_j(\mathcal{U})|S_X).
\end{aligned}$$

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

Corollary B.5 (Interactive Measurability from Rademacher Bound)**.** *Let $\mathcal{U}$ be a class of utility functions and $\mathcal{G}_{\mathcal{U}}$ be the function class defined in equation (B.2). With probability at least $1-\delta$ over the draw of $S \sim D^n$:*

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

Combined with Theorem B.4, this implies:

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

Proof. For $u \in \mathcal{U}$ and $I \in \mathbb{I}[-1,1]$, define $g_{u,I}(X,Y) \coloneqq \langle Y - f(X), \vec{u}(X)\mathbf{1}\left\{v_u(X)\in I\right\}\rangle$. With $\widehat{\text{UC}}(f,u;S)$

defined in Lemma 3.4, it holds by definition that

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

The first inequality is by $\sup_x f(x) - \sup_x g(x) \leq \sup_x |(f-g)(x)|$. Then using (Mohri et al., 2018, Theorem 3.3) (standard symmetrization argument using Rademacher complexity and application of McDiarmid inequality), with probability at least $1-\delta$, it holds that

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

The corollary follows by substituting the bound for $\mathfrak{R}_n(\mathcal{G}_\mathcal{U})$ from Theorem B.4 into the equation above. □

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

Corollary B.6 (Interactive Measurability of $\mathcal{U}_{\mathrm{lin}}$)**.** *The utility calibration error is interactively measurable for the class of linear utilities $\mathcal{U}_{\mathrm{lin}}$ (Example 3.6). The sample complexity is $N = O\left(\frac{C^3\log(n/C)+\log(1/\delta)}{\varepsilon^2}\right)$.*

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

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

Each function in $\mathcal{P}_j(\mathcal{U}_{\mathrm{lin}})$ is a product of a_j and an indicator function $\mathbf{1}\{\langle f(X), a\rangle \in I\}$. First, we consider the class of functions $\mathcal{H}_{a,I} = \{X \mapsto \mathbf{1}\{\langle f(X), a\rangle \in I\} | a \in [-1,1]^C, I \in \mathbb{I}\,[-1,1]\}$. $\mathcal{H}_{a,I}$ is a subclass of the indicator functions of polytopes with two supporting hyperplanes. Thus, $\mathrm{VC}\left(\mathcal{H}_{a,I}\right) = O\left(C\right)$ (Kupavskii, 2020, Theorem 1) and $\mathfrak{R}_n\left(\mathcal{H}_{a,I}\right) = O\left(\sqrt{\frac{C\log(n/C)}{n}}\right)$. We now proceed to bound $\mathfrak{R}_n\left(\mathcal{P}_j\left(\mathcal{U}_{\mathrm{lin}}\right)\right)$. Let $S_X = \{X_1, \ldots, X_n\}$ be n i.i.d.

samples. The empirical Rademacher complexity of $\mathcal{P}_j(\mathcal{U}_{\text{lin}})$ is:

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

Thus, taking expectations over S_X:

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

Using Corollary B.5, the uniform error bound is:

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

□

B.4.1 Sharper interactive measurability for $\mathcal{U}_{\text{lin}}$ and $\mathcal{U}_{\text{rank}}$

The generic Rademacher approach of Theorem B.4 and Corollary B.5 applies broadly. For linear utilities (Example 3.6) and rank-based utilities (Example 3.7), we obtain sharper bounds by analyzing the pseudo-dimension of the underlying real-valued classes, yielding an $O(C\log C)$ upper bound with a log-matching $\Omega(C)$ lower bound. We first recall the definition of pseudo-dimension and some standard results.

Definition B.7. *(Anthony and Bartlett, 2009, pseudo-dimension and Subgraph) For a real-valued class $\mathcal{F}$ of functions on a domain $\mathcal{Z}$, its subgraph class is*

$$\text{Sub}(\mathcal{F}) \;=\; \{\; ((z),r)\mapsto \mathbf{1}\{r\le g(z)\} \;:\; g\in\mathcal{F}\;\}.$$

The pseudo-dimension $\text{Pdim}(\mathcal{F})$ *is the VC dimension of* $\text{Sub}(\mathcal{F})$.

Result 2. *(Goldberg and Jerrum, 1993, Simplified Version of Theorem 2.2) Let $\{h_\theta\}_{\theta\in\mathbb{R}^m}$ be a family of Boolean functions. Suppose that each function h_θ is decided by a Boolean formula of constant size whose atomic predicates are linear inequalities in θ. Then, it holds that*

$$\text{VC}\left(\{h_\theta : \theta\in\mathbb{R}^m\}\right) \;= O\left(m\log m\right).$$

Result 3. *(Mohri et al., 2018, Theorem 10.6) Let $\mathcal{F}$ be a class of functions $h : \mathcal{Z} \to \mathbb{R}$ with range in $[-B, B]$ and $\mathrm{Pdim}(\mathcal{F}) = d$. Then for any $\delta \in (0, 1)$, with probability at least $1 - \delta$ over n i.i.d. samples,*

$$\sup_{h \in \mathcal{F}} \left| \widehat{\mathbb{E}}_n[h] - \mathbb{E}[h] \right| \;\le\; O\left(B \frac{\sqrt{2d \log(n/d)} + \sqrt{\log(1/\delta)}}{\sqrt{n}} \right).$$

We now give upper and lower bounds on the pseudo-dimension for the linear-utility class and then deduce the resulting sample complexity for interactive measurability. Rank-based utilities will follow via a composition corollary at the end of this section.

Proposition B.8 (pseudo-dimension for $\mathcal{U}_{\mathrm{lin}}$)**.** $\mathrm{Pdim}(\mathcal{G}_{\mathrm{lin}}) = \Omega(C)$ *and* $\mathrm{Pdim}(\mathcal{G}_{\mathrm{lin}}) = O\left(C \log C\right)$*, where* $\mathcal{G}_{\mathrm{lin}}$ *is the class of functions induced by* $\mathcal{U}_{\mathrm{lin}}$ *as in Equation* (B.2)*.*

Proof. For $a \in [-1, 1]^C$ and $[s, t] \subseteq [-1, 1]$, define

$$g_{a,s,t}(X, Y) \;=\; \langle Y - f(X),\, a \rangle\, \mathbf{1}\Big\{ s \le \langle f(X), a \rangle \le t \Big\}, \qquad \mathcal{G}_{\mathrm{lin}} = \{ g_{a,s,t} : a \in [-1, 1]^C,\; s \le t \in [-1, 1] \}.$$

Let $p = f(X)$ and $W = Y - f(X)$. By the definition of the subgraph class, for any fixed (X, Y) and threshold $r \in \mathbb{R}$,

$$r \le g_{a,s,t}(X, Y) \iff \big(s \le \langle p, a \rangle \le t \;\wedge\; r \le \langle W, a \rangle \big) \;\vee\; \big(\neg [s \le \langle p, a \rangle \le t] \;\wedge\; r \le 0 \big).$$

We introduce the following atomic linear predicates in the parameters (a, s, t):

$$A_1 :\; \langle p, a \rangle - s \ge 0, \qquad A_2 :\; t - \langle p, a \rangle \ge 0,$$
$$A_3 :\; \langle W, a \rangle - r \ge 0, \qquad A_4 :\; -r \ge 0\,.$$

Note that $(A_1 \wedge A_2) \iff s \le \langle p, a \rangle \le t$, $A_3 \iff r \le \langle W, a \rangle$, and $A_4 \iff r \le 0$. Therefore,

$$r \le g_{a,s,t}(X, Y) \iff \big(A_1 \wedge A_2 \wedge A_3 \big) \;\vee\; \big((\neg A_1 \vee \neg A_2) \wedge A_4 \big).$$

Thus, under the above equivalent, we obtain that $\mathrm{Pdim}\left(\mathcal{G}_{\mathrm{lin}}\right) = O(C \log(C))$, by invoking Proposition 2 with $m = C + 2$.

For the lower bound, fix any $x_0 \in \mathcal{X}$ and write $p \coloneqq f(x_0) \in \Delta^{C-1}$. Relabel coordinates so that $p_C > 0$. Consider the $C - 1$ subgraph points

$$\big((x_0, e_j), 0 \big), \qquad j \in [C - 1].$$

Set $a_C = 0$, $s = -1$, and $t = 1$. Since $p \in \Delta^{C-1}$ and $a \in [-1, 1]^C$, we have $\langle p, a \rangle \in [-1, 1]$, so the interval indicator is always active. Hence, for each $j \in [C - 1]$,

$$g_{a,s,t}(x_0, e_j) = \langle e_j - p, a \rangle = a_j - \sum_{i=1}^{C-1} p_i a_i.$$

Writing $b = (a_1, \ldots, a_{C-1})^\top$ and $q = (p_1, \ldots, p_{C-1})^\top$, the vector of these values is

$$\big(g_{a,s,t}(x_0, e_1), \ldots, g_{a,s,t}(x_0, e_{C-1}) \big)^\top = \big(I_{C-1} - \mathbf{1} q^\top \big) b.$$

Now

$$\det\big(I_{C-1} - \mathbf{1} q^\top \big) = 1 - q^\top \mathbf{1} = 1 - \sum_{i=1}^{C-1} p_i = p_C > 0,$$

so $I_{C-1} - \mathbf{1} q^\top$ is invertible. Let $\ell \in \{0, 1\}^{C-1}$ be any labeling and define $\xi \in \{-1, 1\}^{C-1}$ by

$$\xi_j = \begin{cases} 1, & \ell_j = 1, \\ -1, & \ell_j = 0. \end{cases}$$

Choose

$$b = \delta \big(I_{C-1} - \mathbf{1} q^\top \big)^{-1} \xi$$

with

$$0 < \delta \le \frac{1}{\left\| \left(I_{C-1} - \mathbf{1}q^\top\right)^{-1} \right\|_\infty},$$

so that $\|b\|_\infty \le 1$. Then the corresponding $a \in [-1,1]^C$ is admissible and

$$g_{a,s,t}(x_0, e_j) = \delta \xi_j, \qquad j \in [C-1].$$

Therefore

$$\mathbf{1}\{0 \le g_{a,s,t}(x_0, e_j)\} = \ell_j, \qquad j \in [C-1],$$

so the subgraph class shatters these $C-1$ points. Hence

$$\mathrm{Pdim}(\mathcal{G}_{\mathrm{lin}}) \ge C - 1 = \Omega(C).$$

□

Given the pseudo-dimension bound, we can now directly deduce the sample complexity for interactive measurability.

Corollary B.9 (Sample complexity for $\mathcal{U}_{\mathrm{lin}}$)**.** *Combining Proposition B.8 and Proposition 3, we obtain that*

$$\sup_{u \in \mathcal{U}_{\mathrm{lin}}} \left| \widehat{\mathrm{UC}}(f, u; S) - \mathrm{UC}(f, u) \right| \;\le\; \tilde{O}\left(\sqrt{\frac{C + \log(1/\delta)}{n}} \right).$$

Equivalently, interactive measurability holds with $n = \tilde{O}\big((C + \log(1/\delta))/\varepsilon^2\big)$ for the $\mathcal{U}_{\mathrm{lin}}$ class.

Corollary B.10 (Rank utilities via sorting preprocessing)**.** *Let π_p denote the permutation that sorts $p = f(X)$ in descending order and define the fixed ordering map $T : (p, Y) \mapsto (\tilde{p}, \tilde{Y})$ with $\tilde{p}_s = p_{\pi_p(s)}$ and $\tilde{Y}_s = Y_{\pi_p(s)}$. Then, any rank-based utility may be expressed as a linear utility up to a preprocessing with T. Consequently, $\mathrm{Pdim}(\mathcal{G}_{\mathcal{U}_{\mathrm{rank}}}) = O(C \log C)$ and the sample complexity for interactive measurability for $\mathcal{U}_{\mathrm{rank}}$ matches that of $\mathcal{U}_{\mathrm{lin}}$.*

Proof. Let π_p be the permutation that sorts $p = f(X)$ in descending order and define the fixed ordering map $T : (p, Y) \mapsto (\tilde{p}, \tilde{Y})$ with $\tilde{p}_s = p_{\pi_p(s)}$ and $\tilde{Y}_s = Y_{\pi_p(s)}$. For any $u_\theta \in \mathcal{U}_{\mathrm{rank}}$, we have $[\vec{u}_\theta(p)]_j = \theta_{\mathrm{rank}(p,j)}$ and $v_{u_\theta}(p) = \sum_{s=1}^C \tilde{p}_s \, \theta_s = \langle \tilde{p}, \theta \rangle$. Hence, for any interval $[s,t] \subseteq [-1,1]$,

$$\langle Y - f(X), \vec{u}_\theta(f(X)) \rangle \; \mathbf{1}\{\, s \le v_{u_\theta}(f(X)) \le t \,\} \;=\; \left\langle \tilde{Y} - \tilde{p}, \theta \right\rangle \; \mathbf{1}\{\, s \le \langle \tilde{p}, \theta \rangle \le t \,\},$$

which is exactly the linear-utility template on the transformed sample $T(S)$. As Proposition B.8 is distribution-independent and T is a fixed function, we have $\mathrm{Pdim}(\mathcal{G}_{\mathcal{U}_{\mathrm{rank}}}) = O(C \log C)$. Interactive measurability follows from Proposition 3, similarly to Theorem B.9. □

B.5 Quantile Guarantees

The evaluation methodology in Section 5 relies on plotting the empirical CDF $\widehat{F}_{E,M,n}$, constructed from M utility functions and n data points. $\widehat{F}_{E,M,n}$ serves as a proxy for the true CDF F_E of utility calibration errors. Theorem B.11 bounds the distance between the curves, as characterized by the L_2-distance. While weaker than characterizing the deviation using L_∞-norm, L_2 distance allows a bound without any assumptions on the smoothness of the underlying CDF F_E. The bound depends on two factors: first, the uniform accuracy $\varepsilon_{\mathrm{stat}}$ with which individual utility calibration errors $\mathrm{UC}(f, u)$ can be estimated by $\widehat{\mathrm{UC}}(f, u; S)$, and second, the number of utility functions M sampled to construct the $\widehat{F}_{E,M,n}$.

Theorem B.11. *Let $F_E(e) \coloneqq \mathbb{P}_{u \sim \mathcal{D}_\mathcal{U}}(\mathrm{UC}(f, u) \le e)$ be the true CDF of utility calibration errors, where u is drawn from a distribution $\mathcal{D}_\mathcal{U}$ over the utility class $\mathcal{U}$. Let $\widehat{F}_{E,M,n}(e) \coloneqq \frac{1}{M} \sum_{m=1}^M \mathbf{1}\{\widehat{\mathrm{UC}}(f, u_m; S) \le e\}$ be its empirical estimate, based on M i.i.d. utility functions $u_1, \ldots, u_M \sim \mathcal{D}_\mathcal{U}$ and a data sample S of size n. Assume that, with probability at least $1 - \delta_S$ over the draw of S,*

$$\sup_{u \in \mathcal{U}} |\widehat{\mathrm{UC}}(f, u; S) - \mathrm{UC}(f, u)| \le \varepsilon_{stat}.$$

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

$$\|F_E - \widehat{F}_{E,M,n}\|_{L^2([0,2])} \;\le\; \sqrt{2\varepsilon_{stat}} + \sqrt{\frac{\ln(2/\delta_M)}{M}}.$$

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

We condition on the event (occurring with probability at least $1-\delta_S$) that S is such that $|E(u) - \widehat{E}_S(u)| \le \varepsilon_{\text{stat}}$ for all $u \in \mathcal{U}$. This implies that for any u, $E(u) - \varepsilon_{\text{stat}} \le \widehat{E}_S(u) \le E(u) + \varepsilon_{\text{stat}}$. Consequently, for any $u \in \mathcal{U}$, as $E(u)$ and $\hat{E}(u)$ belong to the set $[0,2]$, it holds for any $e \in [0,2]$ that

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

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

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

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

$$\begin{aligned} \|F_{\widehat{E}_S} - F_E\|^2_{L^2([0,2])} &= \int_0^2 (F_{\widehat{E}_S}(e) - F_E(e))^2 \mathrm{d}e \\ &\le \int_0^2 \Delta_E(e)^2 \mathrm{d}e \;\le\; \int_0^2 \Delta_E(e) \mathrm{d}e, \end{aligned}$$

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

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

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

Next, $\widehat{F}_{E,M,n}(e)$ is the empirical CDF of M i.i.d. samples $\{\widehat{E}_S(u_m)\}_{m=1}^M$ drawn according to $F_{\widehat{E}_S}$. By the Dvoretzky–Kiefer–Wolfowitz (DKW) inequality (Massart, 1990), with probability at least $1-\delta_M$:

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

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

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

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

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

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

B.6 Proof of Proposition 3.2

Proof. At a high-level, the proof is similar to the approach of Rossellini et al. (2025) but some details differ. We start by analyzing the loss function

$$\ell_{\text{util}}(u_Y, \hat{U}; t_0) = (t_0 - u_Y)(\hat{U} - \mathbf{1}\{u_Y \geq t_0\}). \tag{B.6}$$

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

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

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

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 term $\mathbf{1}\{v_u(X) \geq t_0\} - \mathbf{1}\{h(v_u(X)) \geq t_0\}$ equals -1 on E_1 and 1 on E_2, and 0 elsewhere.

$$\begin{aligned}
\Delta R &= \mathbb{E}\Big[(t_0 - u_Y)(-\mathbf{1}\{X \in E_1\}) + (t_0 - u_Y)(\mathbf{1}\{X \in E_2\})\Big] \\
&= \mathbb{E}\Big[(u_Y - t_0)\mathbf{1}\{X \in E_1\}\Big] - \mathbb{E}\Big[(u_Y - t_0)\mathbf{1}\{X \in E_2\}\Big] \\
&= \mathbb{E}\Big[(u_Y - v_u(X) + v_u(X) - t_0)\mathbf{1}\{X \in E_1\}\Big] - \mathbb{E}\Big[(u_Y - v_u(X) + v_u(X) - t_0)\mathbf{1}\{X \in E_2\}\Big] \\
&= \underbrace{\mathbb{E}\Big[(u_Y - v_u(X))\mathbf{1}\{X \in E_1\}\Big]}_{A} + \underbrace{\mathbb{E}\Big[(v_u(X) - t_0)\mathbf{1}\{X \in E_1\}\Big]}_{C} \\
&\quad - \underbrace{\mathbb{E}\Big[(u_Y - v_u(X))\mathbf{1}\{X \in E_2\}\Big]}_{B} - \underbrace{\mathbb{E}\Big[(v_u(X) - t_0)\mathbf{1}\{X \in E_2\}\Big]}_{D}.
\end{aligned}$$

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, $\Delta R = A + C - B - D \leq A - B$. $A - B = \mathbb{E}[(u_Y - v_u(X))\mathbf{1}\{X \in E_1\}] + \mathbb{E}[(v_u(X) - u_Y)\mathbf{1}\{X \in E_2\}]$. 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 $\mathbf{1}\{X \in E_1\}$ and $\mathbf{1}\{X \in E_2\}$ effectively restrict the expectation to regions where $v_u(X)$ falls into certain ranges. By the definition of $\text{UC}(f, u)$, for $E \in \{E_1, E_2\}$ $\mathbb{E}[(u_Y - v_u(X))\mathbf{1}\{X \in E\}] \leq \sup_{I \in \mathbb{I}[-1,1]} |\mathbb{E}[(u_Y - v_u(X))\mathbf{1}\{v_u(X) \in I\}]| \leq \text{UC}(f, u)$.

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 non-decreasing function h, taking the supremum over monotone functions completes the proof. □

B.7 Proof of Proposition 3.3

Proof. The proof is the same as (Rossellini et al., 2025, Lemma A.2.), we include it for completeness. Assume $\text{UC}(f, u) > 0$. Let $U_Y \coloneqq u(f(X), Y)$ denote the realized utility. Both U_Y and $v_u(X)$ take values in $[-1, 1]$.

Let $W \in (0, 2]$ be a chosen bin width. We partition the interval $[-1, 1]$ into $K_W = \lceil 2/W \rceil$ disjoint intervals $A_1, A_2, \ldots, A_{K_W}$. These intervals are constructed as follows: For $j = 1, \ldots, K_W - 1$, let $A_j = [-1 + (j-1)W, -1 + jW)$. For $j = K_W$, let $A_{K_W} = [-1 + (K_W - 1)W, 1]$. This construction ensures that $\bigcup_{j=1}^{K_W} A_j = [-1, 1]$, and each interval A_j has length $\lambda(A_j) \leq W$.

Let $\psi_W : [-1, 1] \to \{1, \ldots, K_W\}$ be the function mapping a value $z \in [-1, 1]$ to the index j of the bin A_j such that $z \in A_j$. We construct a candidate calibrated predictor $g_W(X) \coloneqq \mathbb{E}[U_Y \mid \psi_W(v_u(X))]$. The function

$g_W(X)$ is perfectly calibrated: $\mathbb{E}[U_Y \mid g_W(X)] = \mathbb{E}[\mathbb{E}[U_Y \mid \psi_W(v_u(X))] \mid g_W(X)]$. Since $g_W(X)$ is, by definition, measurable with respect to the sigma-algebra generated by $\psi_W(v_u(X))$, it follows from the properties of conditional expectation that $\mathbb{E}[U_Y \mid g_W(X)] = g_W(X)$ almost surely.

By the definition of $\mathrm{DCU}(f,u)$, which is the infimum of distances $\mathbb{E}|g(X) - v_u(X)|$ over all perfectly calibrated predictors $g(X)$, it holds that $\mathrm{DCU}(f,u) \le \mathbb{E}|g_W(X) - v_u(X)|$. To analyze the right-hand side, we introduce an intermediate term $V_{\mathrm{avgbin}}(X) \coloneqq \mathbb{E}[v_u(X) \mid \psi_W(v_u(X))]$. 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: $\mathbb{E}|g_W(X) - v_u(X)| \le \mathbb{E}|g_W(X) - V_{\mathrm{avgbin}}(X)| + \mathbb{E}|V_{\mathrm{avgbin}}(X) - v_u(X)|$.

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

To bound the first term:

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

By the definition of utility calibration error $\mathrm{UC}(f,u)$, for each bin A_j, we have: $\left|\mathbb{E}\left[(U_Y - v_u(X))\mathbf{1}\left\{v_u(X) \in A_j\right\}\right]\right| \le \sup_{I \in \mathbb{I}[-1,1]} \left|\mathbb{E}\left[(U_Y - v_u(X))\mathbf{1}\left\{v_u(X) \in I\right\}\right]\right| = \mathrm{UC}(f,u)$. Therefore, $\mathbb{E}|g_W(X) - V_{\mathrm{avgbin}}(X)| \le \sum_{j=1}^{K_W} \mathrm{UC}(f,u) = K_W \mathrm{UC}(f,u)$. Since $K_W = \lceil 2/W \rceil$, and for any $x > 0$, $\lceil x \rceil \le x + 1$ (with strict inequality if x is not an integer), we have $K_W \le 2/W + 1$. Thus, $\mathbb{E}|g_W(X) - V_{\mathrm{avgbin}}(X)| \le (2/W + 1)\mathrm{UC}(f,u)$.

Combining the bounds for the two terms, we get: $\mathrm{DCU}(f,u) \le \left(\frac{2}{W} + 1\right)\mathrm{UC}(f,u) + W$. This inequality holds for any chosen bin width $W \in (0,2]$. Our goal is to select W to minimize this upper bound. Set $W = W_{opt} \coloneqq \sqrt{2\mathrm{UC}(f,u)}$, noting that W_{opt} is in the domain $(0,2]$ as $\mathrm{UC}(f,u) \le 2$, for any u and f.

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

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

□

B.8 Proof of Lemma 3.4

Proof. Let $A(X,Y) \coloneqq u(f(X),Y) - v_u(X)$ and $V \coloneqq v_u(X)$. Since $u(\cdot,\cdot), v_u(\cdot) \in [-1,1]$, we have $A(X,Y) \in [-2,2]$ and $V \in [-1,1]$ almost surely. For any interval $I \in \mathbb{I}[-1,1]$, define

$$\begin{aligned}
\mathcal{F}_{\mathrm{int}} &\coloneqq \left\{ h_I : (X,Y) \mapsto A(X,Y)\,\mathbf{1}\{V \in I\} \;\middle|\; I \in \mathbb{I}[-1,1] \right\}, \\
\mathcal{F}_{\le} &\coloneqq \left\{ h_t : (X,Y) \mapsto A(X,Y)\,\mathbf{1}\{V \le t\} \;\middle|\; t \in [-1,1] \right\}.
\end{aligned}$$

By definition,

$$\mathrm{UC}(f,u) = \sup_{I \in \mathbb{I}[-1,1]} \left|\mathbb{E}[h_I]\right|, \qquad \widehat{\mathrm{UC}}(f,u;S) = \sup_{I \in \mathbb{I}[-1,1]} \left|\hat{\mathbb{E}}_n[h_I]\right|,$$

and hence

$$\left|\mathrm{UC}(f,u) - \widehat{\mathrm{UC}}(f,u;S)\right| \;\le\; \sup_{I\in\mathbb{I}[-1,1]} \left|(\mathbb{E}-\hat{\mathbb{E}}_n)h_I\right|.$$

where inequality uses $|\sup a - \sup b| \le \sup|a-b|$ and $||x|-|y|| \le |x-y|$. Under the additional assumption that $V = v_u(X)$ has a continuous distribution, the distinction between closed and half-open interval endpoints is immaterial for both the population quantity and the empirical objective, and the sample values $(V_i)_{i=1}^n$ are pairwise distinct almost surely. For any $a < b$ in $[-1,1]$, we then have $h_{(a,b]} = h_b - h_a$ with $h_t \in \mathcal{F}_{\le}$. Hence, for any linear functional T,

$$\sup_{I\in\mathbb{I}[-1,1]} |T(h_I)| \;\le\; 2 \sup_{t\in[-1,1]} |T(h_t)|.$$

We use the following standard bounded-range uniform deviation bound: for any real-valued function class $\mathcal{H}$ with range $[-M,M]$ and any $\delta \in (0,1)$ (Mohri et al., 2018, Theorem 3.3),

$$\sup_{h\in\mathcal{H}} \left|(\mathbb{E}-\hat{\mathbb{E}}_n)h\right| \;\le\; 2\,\mathfrak{R}_n(\mathcal{H}) \;+\; M\sqrt{\tfrac{\log(1/\delta)}{2n}} \quad \text{w.p. } \ge 1-\delta.$$

Fix the sample $S = \{(X_i,Y_i)\}_{i=1}^n$. Let $V_i := v_u(X_i)$ and $A_i := u(f(X_i),Y_i) - v_u(X_i)$. Let π be a permutation that sorts the V_i: $V_{(1)} \le \cdots \le V_{(n)}$ and set $A_{(i)} := A_{\pi(i)}$. Given S, the sequence $(A_{(i)})_{i=1}^n$ is deterministic. Let $(\sigma_i)_{i=1}^n$ be i.i.d. Rademacher signs, independent of S, and define the filtration

$$\mathcal{F}_k := \sigma(\sigma_1,\ldots,\sigma_k,S), \qquad k = 0,1,\ldots,n.$$

Define partial sums $S_k := \sum_{i=1}^k \sigma_i A_{(i)}$ with $S_0 := 0$. Then $(S_k,\mathcal{F}_k)_{k=0}^n$ is a square-integrable martingale since

$$\mathbb{E}\left[S_k - S_{k-1} \mid \mathcal{F}_{k-1}\right] = A_{(k)}\,\mathbb{E}[\sigma_k \mid \mathcal{F}_{k-1}] = 0, \qquad \mathbb{E}\left[S_n^2 | S\right] = \sum_{i=1}^n A_{(i)}^2.$$

We invoke Doob's L^p submartingale maximal inequality, for $p > 1$,

$$\mathbb{E}\left[\left(\sup_{0\le k\le n} |S_k|\right)^p\right] \;\le\; \left(\tfrac{p}{p-1}\right)^p \mathbb{E}\left[|S_n|^p\right].$$

Taking $p = 2$ yields

$$\mathbb{E}_\sigma\left[\sup_{0\le k\le n} |S_k|^2\right] \;\le\; 4\,\mathbb{E}_\sigma[S_n^2] \;=\; 4\sum_{i=1}^n A_{(i)}^2, \qquad \text{hence} \qquad \mathbb{E}_\sigma\left[\sup_k |S_k|\right] \;\le\; 2\Big(\sum_{i=1}^n A_{(i)}^2\Big)^{1/2}.$$

Then, it holds that

$$\mathfrak{R}_n(\mathcal{F}_{\le} \mid S) \;=\; \frac{1}{n}\,\mathbb{E}_\sigma \max_{0\le k\le n} \sum_{i=1}^k \sigma_i A_{(i)} \;\le\; \frac{1}{n}\,\mathbb{E}_\sigma\left[\sup_k |S_k|\right] \;\le\; \frac{2}{n}\Big(\sum_{i=1}^n A_{(i)}^2\Big)^{1/2} \;\le\; \frac{4}{\sqrt{n}},$$

where the last inequality uses $|A_{(i)}| \le 2$. Then, it holds that

$$\begin{aligned}
\left|\mathrm{UC}(f,u) - \widehat{\mathrm{UC}}(f,u;S)\right| &\le \sup_{I\in\mathbb{I}[-1,1]} \left|(\mathbb{E}-\hat{\mathbb{E}}_n)h_I\right| \\
&\le 2 \sup_{t\in[-1,1]} \left|(\mathbb{E}-\hat{\mathbb{E}}_n)h_t\right| \\
&\le 4\mathfrak{R}_n(\mathcal{F}_{\le}) + 4\sqrt{\tfrac{\log(1/\delta)}{2n}} \\
&\le \frac{16}{\sqrt{n}} + 4\sqrt{\tfrac{\log(1/\delta)}{2n}}.
\end{aligned}$$

For the runtime claim, sorting the values V_i yields a total order, and every admissible interval corresponds to a contiguous block in that sorted order. Thus the empirical objective reduces exactly to the maximum absolute subarray sum of the sorted residuals $(A_{(i)})_{i=1}^n$, which is computed by Kadane scans in linear time after sorting. This gives a runtime of $O(n\log n + nT_{eval})$. □

C Additional Experiments

Appendix C.1 describes datasets, models, and calibration methods. Appendix C.2 takes a deeper look at the empirical behavior of Algorithm 1. To this end, it demonstrates the empirical behavior of Algorithm 1 on two families of utility functions: aligned and misaligned. Finally, Appendix C.4 compiles extended tables and eCDF plots across CIFAR10/100, ImageNet, and an additional text classification experiment. In addition, it introduces two additional utility function classes; one inspired by discounted cumulative gain (DCG) (Järvelin and Kekäläinen, 2002) and the other inspired by semantic-based classification loss (Deng et al., 2012).

C.1 Experimental Details

Experimental Setup: Our experiments were performed using standard image classification benchmarks: CIFAR10, CIFAR100 (Krizhevsky et al., 2009, MIT License), ImageNet-1K (Deng et al., 2009, provided for non-commercial use), and the test split of Yahoo Answers Topics (Zhang et al., 2015, non-commercial use). We used publicly available pretrained models. For each model–dataset combination, the validation data was divided into a calibration set (70%) for training post-hoc methods and a test set (30%) for evaluation. We repeated the experiments across 10 different calibration/test splits and reported the average results. For CIFAR10/100 datasets, we used the model checkpoints available in Chen (2020). For ImageNet-1K, we used `timm` checkpoints (Wightman). For text classification, we used the publicly available model `https://huggingface.co/Koushim/distilbert-yahoo-answers-topic-classifier`, provided under the Apache 2.0 license. All post-hoc methods were trained on the designated calibration set and subsequently evaluated on the test set. All experiments (including hyperparameter tuning) took approximately 120 hours on an A100 80GB NVIDIA GPU.

Calibration Methods and Evaluation: We benchmarked several well-known post-hoc calibration techniques. These include Temperature Scaling (Temp. S.) (Platt et al., 1999; Guo et al., 2017), Vector Scaling (V.S.) (Kull et al., 2017; Guo et al., 2017), Dirichlet recalibration (Kull et al., 2019) with off-diagonal regularization (ODIR), and Isotonic Regression (I.R.) (Zadrozny and Elkan, 2002), applied both globally and in a one-vs-all manner. For these methods, we based our implementation on the publicly available code in (Salvador, 2022). For Dirichlet recalibration, we tuned the regularization parameter over 80 runs using `BayesSearchCV` from `scikit-optimize` (Head et al., 2018) with the goal of minimizing the negative log-likelihood.

Algorithm 1 implementation: Building on the algorithmic template in Algorithm 1, our implementation iteratively (i) audits to find the worst witness over the chosen utility class using the worst-interval estimator, and (ii) applies a masked update along that witness only for points whose predicted utility falls in the identified interval, followed by projection onto the probability simplex. For the stepsize selection, we also employ a backtracking Armijo line search, using JAXOPT (Blondel et al., 2021), to choose a stepsize that decreases the Brier score. The implementation minimizes the utility calibration against the union class $\mathcal{U} = \mathcal{U}_{\text{CWE}} \cup \mathcal{U}_{\text{topK}}$, as defined in Section 3. In addition, at each step, we sampled 264 utility functions from $\mathcal{U}_{\text{rank}}$ and $\mathcal{U}_{\text{lin}}$ to augment the set of utility functions. Hyperparameter tuning was performed using `scikit-optimize` (Head et al., 2018) over the stepsize, the number of steps, and whether to sample additional utility functions from $\mathcal{U}_{\text{rank}}$ and $\mathcal{U}_{\text{lin}}$. For hyperparameter tuning, we performed 20 runs.

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: $\mathcal{U}_{\text{TCE}}$, $\mathcal{U}_{\text{CWE}}$, and $\mathcal{U}_{\text{topK}}$. In addition, we computed other utility calibration metrics for specific utility classes described in Appendix C.4.

C.2 Algorithm 1 for Aligned and Misaligned Utility Classes

To better understand the dynamics of calibrating against different utility structures, this set of experiments focuses on the convergence speed (and final performance) of Algorithm 1 when faced with "aligned" versus "misaligned" non-linear utility classes.

Utility model. Let $R \in [0,1]^{C \times C}$ be a user-defined gain matrix with $R_{ii} = 1$, where R_{ij} is the gain for predicting

class j when the true class is i. For a prediction $p = f(X)$, a rational agent may choose

$$j^\star(p) \;=\; \mathrm{argmax}_{j\in[C]} \sum_{i=1}^{C} p_i \, R_{ij}.$$

As we consider the utility function

$$u_R(p,\, y = e_i) \;=\; R_{i,\, j^\star(p)}\,.$$

Consistent with our framework, we use the associated regressor $v_{u_R}(X) \coloneqq \sum_{i=1}^{C} f(X)_i \, R_{i,\, j^\star(f(X))}$.

We investigate two utility classes to simulate two scenarios: one with all users having the same preferences (aligned) and the other with users having distinct preferences (misaligned).

- Aligned class: models users with a low tolerance for any error. We set $R_{ii} = 1$ and sample all off-diagonal entries ($i \neq j$) i.i.d. from $\mathrm{Unif}(0, 0.1)$.
- Misaligned class: models a mix of specialists with distinct biases. We partition the labels into disjoint subsets $(K_1, K_2, \dots)$. For a user specializing in K_m, we set $R_{ii} = 1$, $R_{ij} = 0.2$ for all $j \in K_m$ with $i \neq j$, and all other off-diagonals to 0. The utility class is a mixture over such specialists.

Evaluation. We illustrate the performance of Algorithm 1 on these classes on CIFAR100 and ImageNet-1K, using ViT and ResNet56 checkpoints. For each utility class, we sampled 512 utility functions per iteration. For each dataset, we track (i) the trajectory of the utility calibration error across iterations of Algorithm 1 for both aligned and misaligned classes (see Figure 2), and (ii) the distribution of per-example utility calibration errors at selected iterations to visualize distributional shifts across the class (see Figure 3). In summary, we provide three panels for each dataset: the iteration curve and two distributional grids (aligned and misaligned).

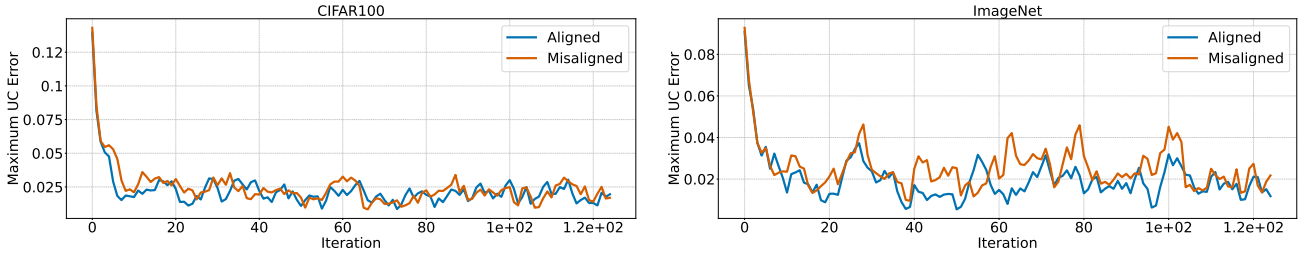

Figure 2: Utility calibration error across iterations for CIFAR100 (left) and ImageNet-1K (right).

Interpretation. Across both datasets and both utility families (aligned and misaligned), the worst-case sampled utility calibration error initially drops and then stabilizes. Nonetheless, utility calibration error distributions shift toward smaller values over iterations. Both aligned and misaligned utility classes displayed similar trends.

C.3 Robustness of eCDF Conclusions to the Sampling Prior

A natural concern is whether the eCDF evaluation methodology's conclusions depend on the choice of sampling distribution $\mathcal{D}_\mathcal{U}$ over the utility class. We address this empirically on `ViT`-ImageNet-1K, comparing the uncalibrated model against the patching algorithm (Algorithm 1).

Experimental design. We vary the *sparsity* of sampled utility vectors as a proxy for different user populations. Each prior is parameterized by $k \in \{50, 70, 100\}$, representing the number of active coordinates in the payoff vector $a \in \partial B_\infty$: for a given k, we select k class indices uniformly at random, assign each a coefficient drawn from $\mathrm{Unif}[-1, 1]$, and set the remaining entries to zero before normalizing to ∂B_∞. For each sparsity level we draw 8 independent instances, yielding 24 priors in total. For each prior we sample $M = 500$ utilities and compute $\widehat{\mathrm{UC}}(f, u)$ for both models.

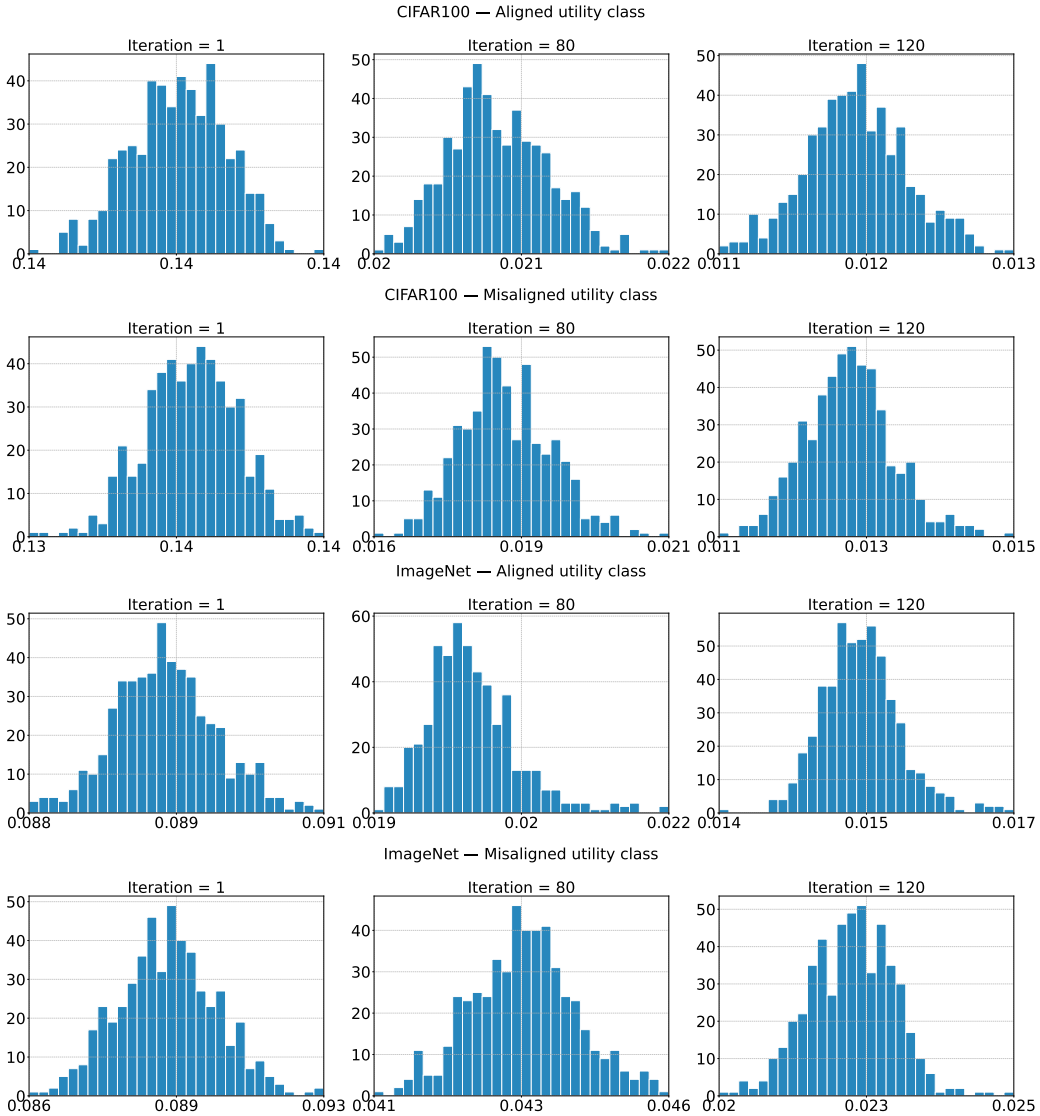

Figure 3: Aligned vs. misaligned utility snapshots across datasets. From top to bottom: CIFAR100 aligned, CIFAR100 misaligned, ImageNet-1K aligned, and ImageNet-1K misaligned, each at selected iterations.

Results. Figure 4 shows the resulting bundle of 24 eCDF curves per model. Despite varying both the sparsity level and the random draw, all curves for the patching algorithm (blue) lie consistently to the left of all curves for the uncalibrated model (red), with no overlap between the two bundles. This confirms that the relative ordering of methods is robust to the choice of prior: the patching algorithm reduces utility calibration errors uniformly across the utility class regardless of the sampling prior used.

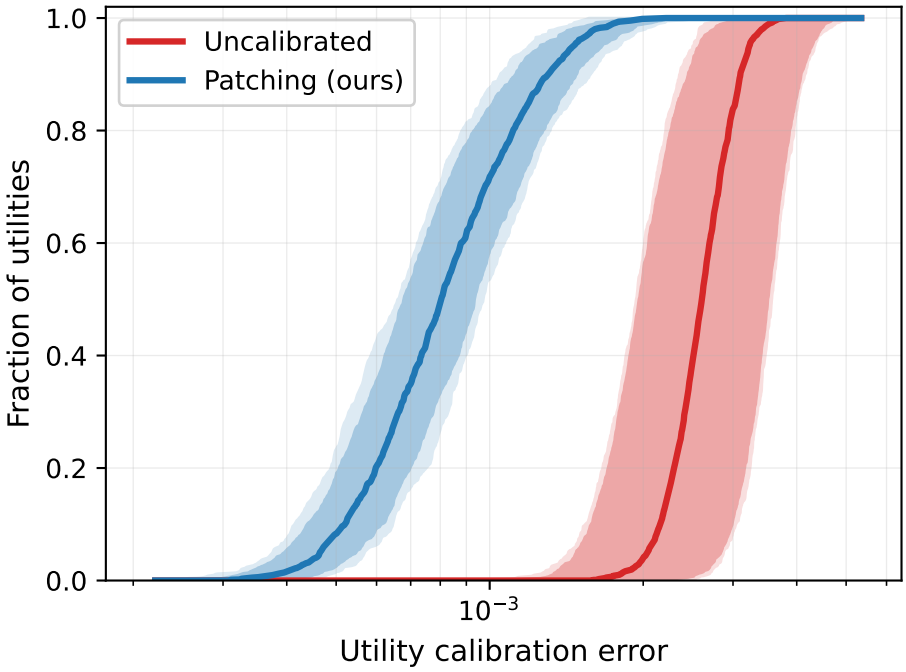

Figure 4: eCDF bundle for `ViT`-ImageNet-1K under 24 different sampling priors (sparsity $k \in \{50, 70, 100\}$, 8 draws each, M=500 utilities per prior). Shaded bands show the min–max range (light) and interquartile range (dark); bold lines show the median. The two bundles do not overlap, confirming that the dominance of the patching algorithm (blue) over the uncalibrated model (red) is robust to the choice of prior.

C.4 Extended Results and Additional Modalities

We begin by introducing two additional utility families below:

Discounted cumulative gain (DCG). We introduce the following discounted cumulative gain utility class, inspired by graded relevance evaluation (Järvelin and Kekäläinen, 2002). We note that this class highly resembles rank-based and topK utilities in Example 3.7. Let $\gamma \in \mathbb{R}_+$ and define the discount sequence $\theta_r^{(\gamma)} = (\log_2(1+r))^{-\gamma}$ for ranks $r \in [C]$. For a prediction $p = f(X)$ and outcome $Y = e_j$, the realized gain is $u_\gamma(p, e_j) = \theta^{(\gamma)}_{\mathrm{rank}(p,j)}$ with associated regressor $v_{u_\gamma}(X) = \sum_i f(X)_i \, \theta^{(\gamma)}_{\mathrm{rank}(f(X),i)}$. Finally, for simplicity of evaluation, we used a fixed grid for γ and thus define the class $\mathcal{U}_{\mathrm{dcg}} \coloneqq \{u_\gamma \mid \gamma \in \{0.5, 0.75, 1, 1.25, 1.5, 2\}\}$.

Semantic similarity utilities. For ImageNet, we construct class embeddings $E \in \mathbb{R}^{C \times d}$ using the pre-trained Google News Word2Vec model (`word2vec-google-news-300` loaded via `gensim`) (Mikolov et al., 2013; Rehurek and Sojka, 2011). Class names are tokenized by lowercasing, replacing underscores/hyphens with spaces, and keeping alphanumeric tokens. For class i, a class embedding E_i is the average of in-vocabulary token vectors. We compute the cosine similarity matrix $S \in [-1, 1]^{C \times C}$ such that $S_{i,j} = 1$ if $i = j$ and $S_{i,j}$ is the cosine similarity between E_i and E_j. Finally, we define the utility by expected similarity: $u_{\mathrm{w2v}}(p, e_j) = \sum_i p_i \, S[i, j]$. We denote this single-utility class by $\mathcal{U}_{\mathrm{w2v}}$.

The results are presented in tables and figures as follows. We note two general observations that illustrate the usefulness of our evaluation framework. (1) The order of performance among different post-hoc methods

Table 2: `RepVGG-CIFAR10` calibration results, comparing post-hoc methods using Accuracy, Brier Score, binned ECEs (TCE^{Bin}, CWE^{Bin}), and utility calibration errors: $\mathcal{U}_{\text{CWE}}$, $\mathcal{U}_{\text{topK}}$, and $\mathcal{U}_{\text{dcg}}$, with mean ± std.

Method	Accuracy ($\times 10^2$)	Brier Score ($\times 10^2$)	CWE^{bin} ($\times 10^4$)	TCE^{bin} ($\times 10^3$)	$\mathcal{U}_{\text{CWE}}$ ($\times 10^4$)	$\mathcal{U}_{\text{topK}}$ ($\times 10^3$)	$\mathcal{U}_{\text{dcg}}$ ($\times 10^3$)
Uncalibrated	94.5 ± 0.474	8.92 ± 0.774	39.6 ± 2.64	37.3 ± 4.5	106.0 ± 7.77	39.0 ± 4.24	24.7 ± 3.05
Dirichlet	94.7 ± 0.468	7.95 ± 0.629	47.0 ± 4.28	17.6 ± 2.5	89.90 ± 4	18.5 ± 1	12.8 ± 1
IR	94.6 ± 0.584	8.12 ± 0.62	24.9 ± 2.03	9.05 ± 3.39	76.90 ± 4	15.4 ± 0.6	10.5 ± 0.5
Temp. Scaling	94.5 ± 0.474	8.23 ± 0.669	44.2 ± 0.777	19.6 ± 3.23	89.70 ± 5	22.8 ± 1	14.0 ± 1.0
Vector Scaling	94.7 ± 0.47	7.98 ± 0.626	47.6 ± 4.52	17.0 ± 2.37	95.30 ± 5	16.1 ± 0.9	9.5 ± 0.8
Patching	94.7 ± 0.502	8.08 ± 0.607	35.3 ± 5.03	10.5 ± 2.88	73.3 ± 8.07	10.0 ± 2.1	4.71 ± 2.26

Table 3: `ResNet-CIFAR10` calibration results, comparing post-hoc methods using Accuracy, Brier Score, binned ECEs (TCE^{Bin}, CWE^{Bin}), and utility calibration errors: $\mathcal{U}_{\text{CWE}}$, $\mathcal{U}_{\text{topK}}$, and $\mathcal{U}_{\text{dcg}}$, with mean ± std.

Method	Accuracy ($\times 10^2$)	Brier Score ($\times 10^2$)	CWE^{bin} ($\times 10^4$)	TCE^{bin} ($\times 10^3$)	$\mathcal{U}_{\text{CWE}}$ ($\times 10^4$)	$\mathcal{U}_{\text{topK}}$ ($\times 10^3$)	$\mathcal{U}_{\text{dcg}}$ ($\times 10^3$)
Uncalibrated	93.9 ± 0.155	10.1 ± 0.341	48.2 ± 1.87	40.5 ± 1.4	120.0 ± 9.73	41.3 ± 2.08	27.3 ± 0.986
Dirichlet	94.1 ± 0.253	9.02 ± 0.327	37.8 ± 6.1	14.4 ± 2.13	85.80 ± 6	14.2 ± 1	8.4 ± 1
IR	94.1 ± 0.151	9.2 ± 0.312	28.8 ± 2.99	10.2 ± 2.73	77.90 ± 5	16.2 ± 0.2	11.8 ± 0.5
Temp. Scaling	93.9 ± 0.155	9.29 ± 0.336	46.9 ± 1.42	14.2 ± 1.76	92.90 ± 5	17.1 ± 1.0	9.2 ± 1
Vector Scaling	94.1 ± 0.313	9.11 ± 0.367	41.9 ± 1.69	13.1 ± 1.23	84.70 ± 4	13.1 ± 0.5	6.3 ± 0.8
Patching	94.0 ± 0.284	9.2 ± 0.321	42.4 ± 4.27	10.4 ± 1.14	79.2 ± 13.0	11.0 ± 0.88	4.86 ± 1.2

varies with respect to the evaluated utility class, stressing the need to account for downstream usage when evaluating calibration. (2) While post-hoc methods generally improve various metrics, eCDF plots reveal additional information. For instance, in multiple settings, post-hoc methods actively worsened (shifted to the right) the distribution of utility calibration errors.

Metrics	Results given in			
	CIFAR10	CIFAR100	ImageNet	Yahoo
TCE^{bin}, CWE^{bin}, $\mathcal{U}_{\text{CWE}}$, $\mathcal{U}_{\text{topK}}$, $\mathcal{U}_{\text{dcg}}$, $\mathcal{U}_{\text{w2v}}$	Tables 2 and 3	Tables 4 and 5	Tables 6 and 7	Table 8
$\mathcal{U}_{\text{lin}}$ and $\mathcal{U}_{\text{rank}}$	Figure 5	Figure 6	Figure 7	Figure 8

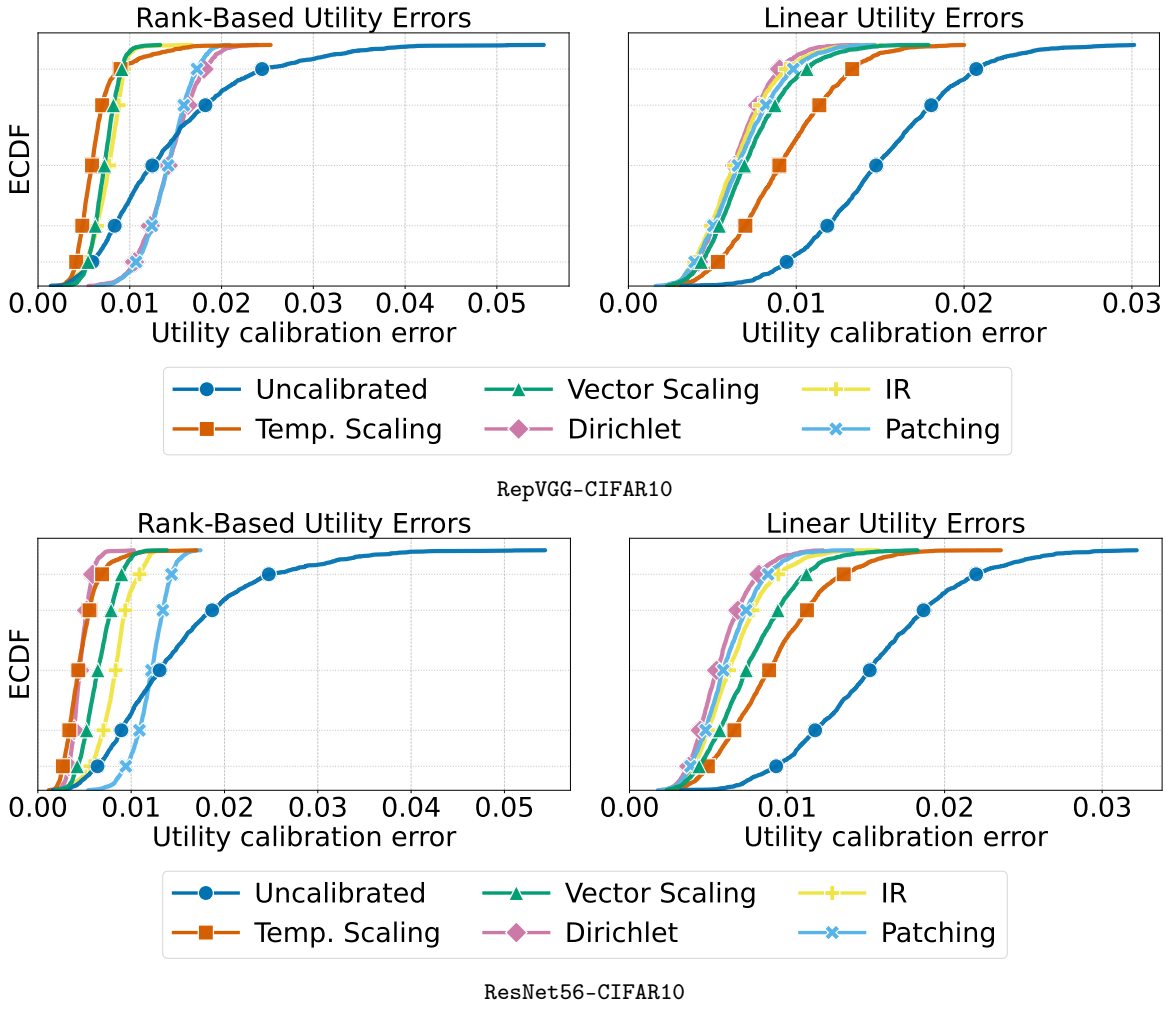

Figure 5: CIFAR10 eCDF plots for $\mathcal{U}_{\text{lin}}$ and $\mathcal{U}_{\text{rank}}$.

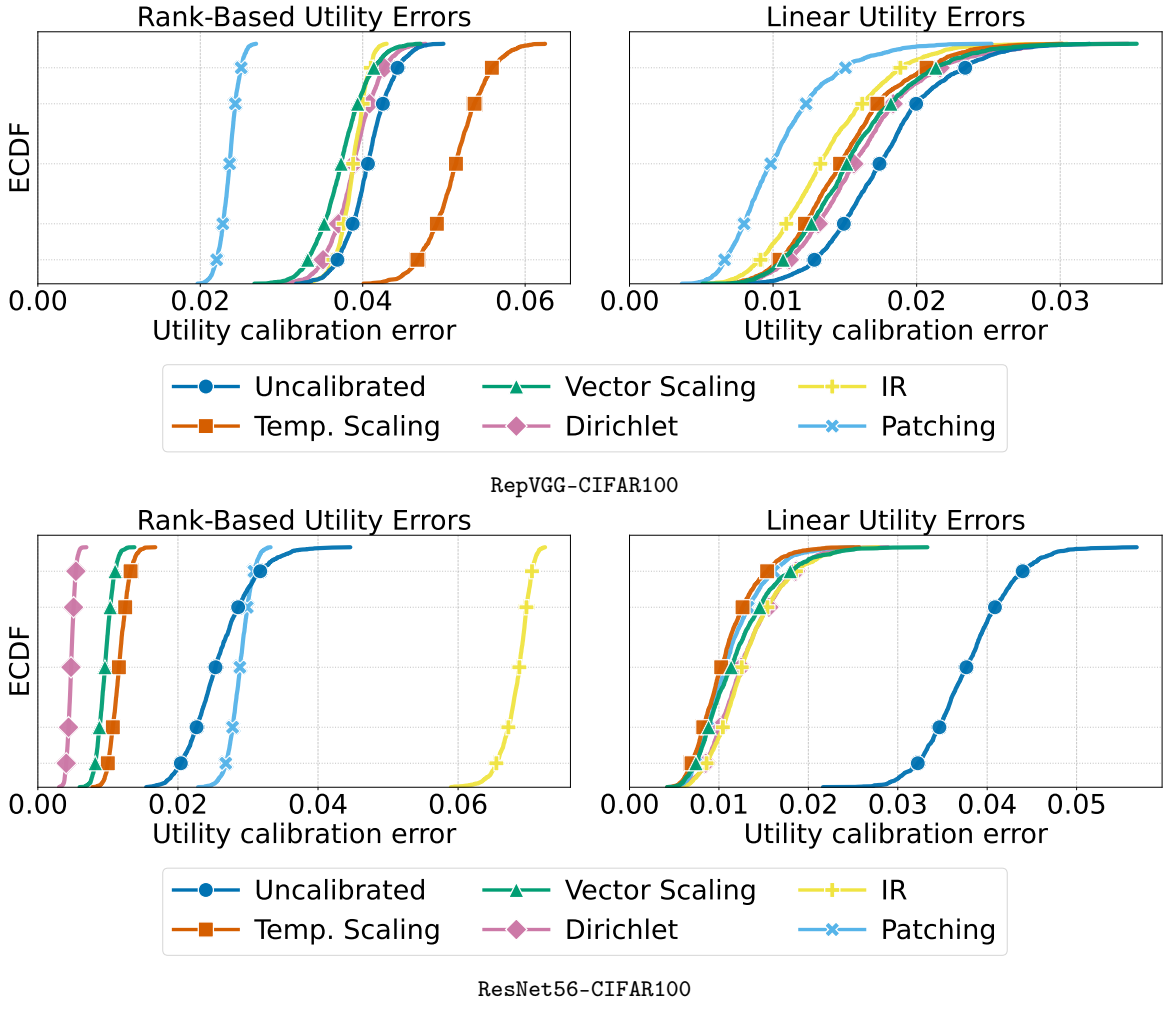

Figure 6: CIFAR100 eCDF plots for $\mathcal{U}_{\text{lin}}$ and $\mathcal{U}_{\text{rank}}$.

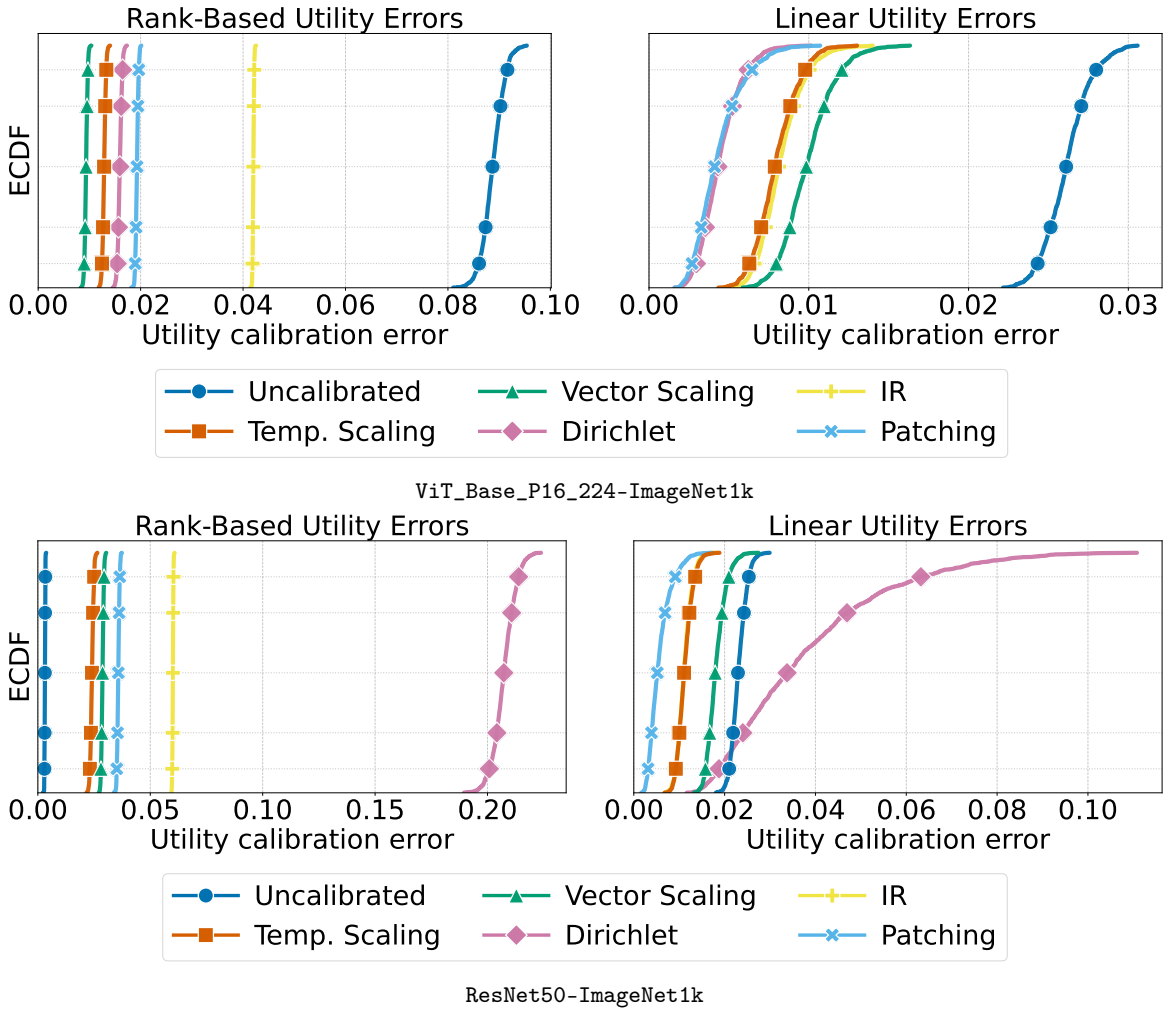

Figure 7: ImageNet eCDF plots for $\mathcal{U}_{\text{lin}}$ and $\mathcal{U}_{\text{rank}}$.

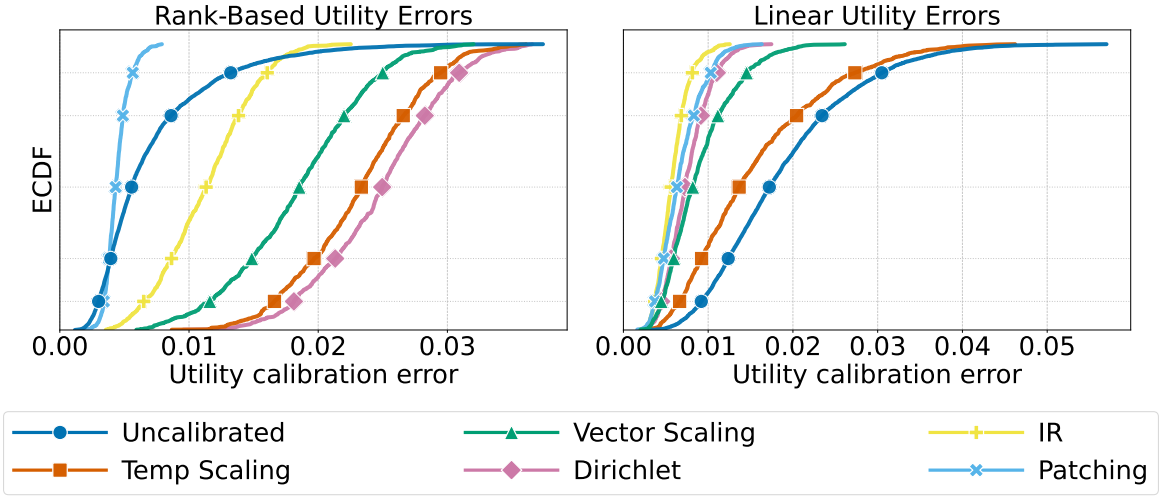

Figure 8: Yahoo Answers eCDF plots for $\mathcal{U}_{\text{lin}}$ and $\mathcal{U}_{\text{rank}}$.

Table 4: `RepVGG-CIFAR100` calibration results, comparing post-hoc methods using Accuracy, Brier Score, binned ECEs (TCE^{Bin}, CWE^{Bin}), and utility calibration errors: $\mathcal{U}_{\text{CWE}}$ and $\mathcal{U}_{\text{topK}}$, with mean ± std.

Method	Accuracy ($\times 10^2$)	Brier Score ($\times 10^2$)	CWE^{bin} ($\times 10^4$)	TCE^{bin} ($\times 10^3$)	$\mathcal{U}_{\text{CWE}}$ ($\times 10^4$)	$\mathcal{U}_{\text{topK}}$ ($\times 10^3$)
Uncalibrated	77.2 ± 0.481	32.7 ± 0.867	15.2 ± 0.646	58.1 ± 7.41	72.8 ± 4.91	55.6 ± 0.77
Dirichlet	76.5 ± 0.806	32.9 ± 1.03	16.9 ± 1.41	49.1 ± 5.86	78.00 ± 4	51.1 ± 1
IR	76.6 ± 0.652	32.8 ± 1.09	14.4 ± 0.605	30.4 ± 4.66	66.20 ± 3	49.5 ± 2
Temp. Scaling	77.2 ± 0.481	32.5 ± 0.87	16.2 ± 0.532	49.1 ± 7.67	76.00 ± 2	65.9 ± 2
Vector Scaling	76.8 ± 0.691	33.0 ± 1.07	17.8 ± 0.835	50.5 ± 4.55	75.00 ± 3	51.2 ± 2
Patching	77.1 ± 0.493	32.0 ± 0.801	18.4 ± 0.656	24.1 ± 4.19	65.3 ± 4.79	24.2 ± 2.01

Table 5: `ResNet-CIFAR100` calibration results, comparing post-hoc methods using Accuracy, Brier Score, binned ECEs (TCE^{Bin}, CWE^{Bin}), and utility calibration errors: $\mathcal{U}_{\text{CWE}}$, $\mathcal{U}_{\text{topK}}$, and $\mathcal{U}_{\text{dcg}}$, with mean ± std.

Method	Accuracy ($\times 10^2$)	Brier Score ($\times 10^2$)	CWE^{bin} ($\times 10^4$)	TCE^{bin} ($\times 10^3$)	$\mathcal{U}_{\text{CWE}}$ ($\times 10^4$)	$\mathcal{U}_{\text{topK}}$ ($\times 10^3$)	$\mathcal{U}_{\text{dcg}}$ ($\times 10^3$)
Uncalibrated	72.8 ± 0.455	41.6 ± 0.681	14.9 ± 0.778	140.0 ± 4.67	80.1 ± 3.94	143.0 ± 6.92	114.0 ± 4.07
Dirichlet	71.9 ± 0.546	38.8 ± 0.759	15.9 ± 0.935	48.5 ± 12.9	74.00 ± 2	61.2 ± 3	50.4 ± 3
IR	72.2 ± 0.376	39.5 ± 0.706	15.4 ± 1.23	34.3 ± 3.37	70.70 ± 2	71.2 ± 2	51.1 ± 2
Temp. Scaling	72.8 ± 0.455	38.3 ± 0.523	15.0 ± 0.424	30.3 ± 4.84	67.20 ± 2	29.5 ± 2	15.8 ± 2
Vector Scaling	72.2 ± 0.508	38.8 ± 0.685	18.1 ± 0.845	31.4 ± 5.42	76.00 ± 3	36.1 ± 3	22.6 ± 3
Patching	71.8 ± 0.361	38.4 ± 0.494	21.8 ± 0.388	20.2 ± 3.51	68.6 ± 7.0	27.1 ± 7.17	8.55 ± 1.94

Table 6: `ResNet-ImageNet` calibration results, comparing post-hoc methods using Accuracy, Brier Score, binned ECEs (TCE^{Bin}, CWE^{Bin}), and utility calibration errors: $\mathcal{U}_{\text{CWE}}$, $\mathcal{U}_{\text{topK}}$, $\mathcal{U}_{\text{dcg}}$, and $\mathcal{U}_{\text{w2v}}$, with mean ± std.

Method	Accuracy ($\times 10^2$)	Brier Score ($\times 10^2$)	CWE^{bin} ($\times 10^4$)	TCE^{bin} ($\times 10^3$)	$\mathcal{U}_{\text{CWE}}$ ($\times 10^4$)	$\mathcal{U}_{\text{topK}}$ ($\times 10^3$)	$\mathcal{U}_{\text{dcg}}$ ($\times 10^3$)	$\mathcal{U}_{\text{w2v}}$ ($\times 10^3$)
Uncalibrated	80.4 ± 0.34	29.7 ± 0.593	1.42 ± 0.0411	87.0 ± 2.29	31.4 ± 1.23	91.0 ± 1.88	62.7 ± 1.89	43.1 ± 1.48
Dirichlet	44.2 ± 25.2	66.6 ± 27.2	10.9 ± 5.1	99.0 ± 46.8	1860.00 ± 580	990.0 ± 3	429.0 ± 59	59.9 ± 38.5
IR	80.0 ± 0.313	29.7 ± 0.49	1.45 ± 0.0261	39.3 ± 2.6	30.50 ± 0.6	65.2 ± 1	47.3 ± 0.6	37.5 ± 1.29
Temp. Scaling	80.4 ± 0.34	28.7 ± 0.561	1.59 ± 0.0308	51.0 ± 3.17	31.50 ± 0.7	59.3 ± 1	24.7 ± 0.7	20.0 ± 1.0
Vector Scaling	79.7 ± 0.312	30.7 ± 0.424	1.9 ± 0.0329	69.6 ± 2.31	31.60 ± 0.8	67.1 ± 2	44.4 ± 1	32.5 ± 2.44
Patching	80.2 ± 0.301	28.1 ± 0.529	2.15 ± 0.105	20.2 ± 2.73	32.1 ± 2.33	40.1 ± 9.95	12.9 ± 7.05	18.4 ± 6.11

Table 7: `ViT-ImageNet` calibration results, comparing post-hoc methods using Accuracy, Brier Score, binned ECEs (TCE^{Bin}, CWE^{Bin}), and utility calibration errors: $\mathcal{U}_{\text{CWE}}$, $\mathcal{U}_{\text{topK}}$, $\mathcal{U}_{\text{dcg}}$, and $\mathcal{U}_{\text{w2v}}$, with mean ± std.

Method	Accuracy ($\times 10^2$)	Brier Score ($\times 10^2$)	CWE^{bin} ($\times 10^4$)	TCE^{bin} ($\times 10^3$)	$\mathcal{U}_{\text{CWE}}$ ($\times 10^4$)	$\mathcal{U}_{\text{topK}}$ ($\times 10^3$)	$\mathcal{U}_{\text{dcg}}$ ($\times 10^3$)	$\mathcal{U}_{\text{w2v}}$ ($\times 10^3$)
Uncalibrated	85.2 ± 0.297	22.6 ± 0.324	2.46 ± 0.0237	94.2 ± 2.27	30.4 ± 0.976	124.0 ± 1.03	101.0 ± 1.56	68.8 ± 0.802
Dirichlet	85.4 ± 0.191	21.3 ± 0.361	1.34 ± 0.0438	13.7 ± 1.53	27.00 ± 0.5	26.1 ± 0.7	6.4 ± 0.4	7.52 ± 0.926
IR	84.9 ± 0.276	22.9 ± 0.371	1.1 ± 0.0183	33.1 ± 1.78	27.40 ± 0.5	54.1 ± 0.9	41.6 ± 0.5	38.1 ± 1.69
Temp. Scaling	85.2 ± 0.297	21.8 ± 0.398	1.26 ± 0.0239	30.0 ± 2.85	28.10 ± 0.6	45.2 ± 0.5	20.8 ± 0.6	11.7 ± 1.0
Vector Scaling	84.8 ± 0.291	22.8 ± 0.376	1.54 ± 0.0442	35.1 ± 2.83	28.30 ± 0.3	37.4 ± 2	21.9 ± 1	13.1 ± 0.811
Patching	85.2 ± 0.258	21.6 ± 0.347	1.56 ± 0.0483	10.3 ± 2.28	28.9 ± 0.816	19.4 ± 2.85	5.13 ± 1.48	11.8 ± 0.608

Table 8: Yahoo calibration results, comparing post-hoc methods using Accuracy, Brier Score, binned ECEs (TCE^{Bin}, CWE^{Bin}), and utility calibration errors: $\mathcal{U}_{\text{CWE}}$, $\mathcal{U}_{\text{topK}}$, and $\mathcal{U}_{\text{dcg}}$, with mean ± std.

Method	Accuracy ($\times 10^2$)	Brier Score ($\times 10^2$)	CWE^{bin} ($\times 10^4$)	TCE^{bin} ($\times 10^3$)	$\mathcal{U}_{\text{CWE}}$ ($\times 10^4$)	$\mathcal{U}_{\text{topK}}$ ($\times 10^3$)	$\mathcal{U}_{\text{dcg}}$ ($\times 10^3$)
Uncalibrated	72.4 ± 0.377	39.0 ± 0.433	100.0 ± 1.84	31.3 ± 2.2	213.0 ± 19.0	33.2 ± 5.27	19.9 ± 1.38
Dirichlet	72.9 ± 0.375	38.4 ± 0.414	67.6 ± 3.88	11.1 ± 1.84	158.00 ± 4	33.6 ± 2	12.2 ± 2
IR	72.8 ± 0.381	38.6 ± 0.425	38.1 ± 4.22	10.1 ± 1.06	150.00 ± 5	30.8 ± 0.7	11.9 ± 0.4
Temp. Scaling	72.4 ± 0.377	38.9 ± 0.429	95.9 ± 1.39	18.8 ± 2.17	213.00 ± 6	31.4 ± 2	9.5 ± 1
Vector Scaling	72.7 ± 0.37	38.6 ± 0.457	80.1 ± 4.59	13.3 ± 2.11	169.00 ± 7	31.2 ± 2	9.1 ± 1
Patching	72.7 ± 0.365	38.8 ± 0.419	72.7 ± 2.9	12.9 ± 1.38	172.0 ± 6.0	21.7 ± 3.42	4.57 ± 1.11