HIERARCHICAL UTILITY CALIBRATION FOR STRUCTURED MULTICLASS DECISIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

In multiclass probabilistic prediction, Utility Calibration (UC), which focuses auditing on specified utilities, has recently received attention as a way to guarantee downstream decisions while controlling computational and sample requirements. At the same time, some multiclass problems have meaningful label hierarchies that play important roles in medicine and image classification, yet how UC evaluates utility within a hierarchy remains insufficiently understood. We show that the difference between realized utility and predicted mean utility admits an exact decomposition into a sum of contributions from the internal nodes of the label tree. This decomposition shows that positive and negative contributions from different nodes can cancel, and that even when UC is small, the utility errors remaining in parts of the hierarchy need not be small. To address this problem, we propose Hierarchical Utility Calibration (HUC), which evaluates each node contribution before summation while retaining the same target utility, subgroup, and predicted-utility interval. We further provide finite-sample evaluation over all predicted-utility intervals and propose HUC-Boost, which updates only violated internal nodes, with theoretical guarantees for both.

1 INTRODUCTION

Calibration of probabilistic predictions is a basic requirement for using predicted probabilities in decision-making, risk assessment, and communication with human users. For multiclass prediction, prior work has studied marginal and class-wise calibration, which evaluate individual components, as well as multiclass recalibration methods that transform the entire probability vector (Guo et al., 2017; Vaicenavicius et al., 2019; Kull et al., 2019; Fujisawa & Futami, 2025). The resulting guarantees and the difficulty of evaluation depend on what is conditioned on and which directions of prediction error are examined. Canonical calibration, a strong calibration condition, requires the true label distribution given the complete prediction vector to agree with that vector and therefore allows the vector to be reused post hoc for a broad range of decisions. Yet Gopalan et al. (2024) showed that testing it without assumptions requires exponentially many samples.

When the goal is to guarantee only prespecified decisions rather than every possible future use, it is unnecessary to evaluate every direction of the complete prediction vector. Utility Calibration (UC) has recently received attention as a way to focus the audit on specified utilities (Hegazy et al., 2026; Rossellini et al., 2025). UC evaluates, over intervals of predicted mean utility, the agreement between the utility realized in the data and the mean utility that the model computes under the assumption that its predictive distribution is correct. It thereby focuses the audit on decisions that will actually be used, rather than on the entire high-dimensional prediction vector. To guarantee several decisions simultaneously, one includes the corresponding utilities in a utility class.

Multiclass labels, however, may themselves have a semantic hierarchy. In medicine, disease groups, clinical states, and severity levels may be distinguished along a tree (Yu et al., 2025); in image classification, prediction may proceed from coarse to fine-grained categories (Deng et al., 2009; Van Horn et al., 2018). Such predictions need not always traverse the tree all the way to a leaf and return a single final class (Goren et al., 2024; Mortier et al., 2022). A coarse decision may use only an upper-level branch, whereas another decision may further inspect only a particular subtree (Goren et al., 2024; Plaud et al., 2025; Mortier et al., 2022). In such hierarchical problems, it has been unclear how UC aggregates errors on the tree into a single utility residual.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

We show for the first time that the difference between realized utility and predicted mean utility can be decomposed exactly into a sum of contributions from the internal nodes of the label tree. Because UC evaluates the residual after adding these contributions, positive and negative node contributions can cancel. Consequently, even when UC is small, nonzero utility error may remain in some part of the hierarchy, and UC may fail to identify the branch that should be inspected or corrected. Section 2.2 demonstrates this phenomenon on a four-leaf tree.

This issue is not resolved by evaluating each leaf separately or by uniformly auditing the branch probabilities of all internal nodes. Calibration of individual leaves does not, in general, determine branch probabilities conditional on having reached the parent subtree. Conversely, auditing every branch either uses a different target population at each node or includes branches that do not affect the target utility. What is needed is to distinguish the node contributions that constitute the original residual without changing the target utility, subgroup, or predicted-utility interval.

We therefore propose Hierarchical Utility Calibration (HUC), which evaluates each node contribution before aggregation. HUC uses the same target utility, subgroup, and predicted-utility interval at every node, allowing node contributions to the utility residual to be compared within the same target population. Moreover, the local utility contrast is zero at nodes that do not affect the target utility, so such nodes are excluded automatically. We also provide a finite-sample estimator of HUC with a theoretical guarantee uniform over all predicted-utility intervals. We then propose HUC-Boost, a post-processing recalibration algorithm that updates only violated internal nodes, and establish preservation of the probabilistic structure, decreasing log loss, and finite termination at the population level. Our main contributions are as follows.

1. **Tree decomposition and HUC.** We show that the utility residual admits an exact decomposition into a sum of internal-node contributions (Section 2.2 and Theorem 3.1) and propose HUC based on this decomposition (Definition 3.2).
2. **HUC estimation and correction.** We give a finite-sample estimator of HUC over all predicted-utility intervals (Section 3.1) and propose HUC-Boost, a post-hoc recalibration algorithm (Algorithm 1) with a theoretical guarantee (Theorem 3.5).
3. **Systematic empirical evaluation.** We evaluate HUC-Boost across datasets with different modalities and hierarchical structures, and systematically compare direct UC-Boost and HUC-Boost, parametric recalibration methods, and two-stage combinations of the parametric methods with UC-Boost or HUC-Boost (Section 5 and Appendices J and K).

2 Preliminaries and a Limitation of Utility Calibration

2.1 Multiclass Calibration and Utility Calibration

Calibration asks whether predicted probabilities agree with realized frequencies among observations that receive the same prediction. Let the input be $X \in \mathcal{X}$, the label be $Y \in \mathcal{Y} := \{1, \dots, K\}$, and the probability simplex be $\Delta^{K-1} := \{p \in [0,1]^K : \sum_{y \in \mathcal{Y}} p_y = 1\}$. Let the predictor be $f : \mathcal{X} \to \Delta^{K-1}$ and write its output as $P := f(X)$. Unless otherwise specified, $\mathbb{P}$ and $\mathbb{E}$ denote probability and expectation under the data-generating distribution, respectively. For label y, let $e_y \in \mathbb{R}^K$ denote the corresponding standard basis vector. The fundamental residual for multiclass probabilistic prediction is $e_Y - P$. The strength of the calibration guarantee and the difficulty of auditing depend on what is conditioned on and which directions of this vector residual are inspected.

Definition 2.1 (Canonical calibration)**.** *A predictor f is canonically calibrated if $\mathbb{E}[e_Y \mid P] = P$ holds almost surely (Gopalan et al., 2024).*

We adopt an audit-moment formulation of calibration, following the weighted-calibration viewpoint of Gopalan et al. (2024). For later use, we also include input-dependent subgroup weights, in the spirit of multicalibration (Hebert-Johnson et al., 2018). Let $\mathcal{C} \subset \{c : \mathcal{X} \to [-1,1]\}$ be a class of measurable subgroup weights and $\mathcal{W} \subset \{w : \Delta^{K-1} \to [-1,1]^K\}$ a class of measurable prediction-dependent audit directions. For $c \in \mathcal{C}$ and $w \in \mathcal{W}$, define

$$M_{c,w}(f) := \mathbb{E}[c(X)\,\langle w(P), e_Y - P\rangle] \tag{1}$$

For $c \equiv 1$, vanishing moments for all measurable $w : \Delta^{K-1} \to [-1,1]^K$ characterize canonical calibration; nonconstant c extends the audit to input-defined subgroups. We call an expected residual of

this form an *audit moment*, and a function that multiplies the residual an *audit function*. Given pre-specified classes $\mathcal{C}$ and $\mathcal{W}$ of audit functions, our goal is to make these audit moments simultaneously small. This naturally leads us to consider the worst-case audit moment $\sup_{c\in\mathcal{C},w\in\mathcal{W}} |M_{c,w}(f)|$.

Approaching canonical calibration requires a sufficiently rich family of audit directions that depend on the complete prediction vector. However, Gopalan et al. (2024) showed that testing canonical calibration without additional assumptions requires a number of samples exponential in K. For a few known downstream decisions, it is more direct to restrict auditing to the quantities they require.

Utility Calibration formalizes this idea. Following Hegazy et al. (2026), we define a *utility* as a bounded measurable function

$$u : \Delta^{K-1} \times \mathcal{Y} \to [-1, 1]. \tag{2}$$

It represents the final utility obtained when outcome y is realized after prediction p has been used, and may already incorporate an action-selection or output rule. For example, let the finite action set be $\mathcal{A}$, let the realized utility of action $a \in \mathcal{A}$ be $U(a, y) \in [-1, 1]$, and let the prediction-based action rule be $\psi : \Delta^{K-1} \to \mathcal{A}$. Then $u_\psi(p, y) := U(\psi(p), y)$ is a utility of the form in Eq. (2). We write $\mathcal{U}$ for a collection of utilities to be guaranteed simultaneously and call it a utility class. We specify this class before observing the audit sample, using the target actions or action comparisons.

For $p \in \Delta^{K-1}$, let $\widetilde{Y} \sim p$ be a virtual label drawn from p. Define the *predicted mean utility* by

$$\mu_u(p) := \mathbb{E}_{\widetilde{Y}\sim p}[u(p, \widetilde{Y})] = \sum_{y\in\mathcal{Y}} p_y u(p, y) \tag{3}$$

Because u takes values in $[-1, 1]$, we have $\mu_u(p) \in [-1, 1]$. Under canonical calibration, the conditional mean of the realized utility agrees with $\mu_u(P)$. Unlike canonical calibration, UC focuses auditing on the specified utility class by ranging over intervals of predicted mean utility rather than over all directions of the complete prediction vector. Let $\mathcal{I} := \{I \subseteq [-1, 1] : I \text{ is an interval}\} \cup \{\varnothing\}$.

Definition 2.2 (Utility Calibration)**.** *The Utility Calibration error with respect to a subgroup class $\mathcal{C}$ and utility class $\mathcal{U}$ is*

$$\mathrm{UC}(f; \mathcal{C}, \mathcal{U}) := \sup_{c\in\mathcal{C},\, u\in\mathcal{U},\, I\in\mathcal{I}} |\mathbb{E}[c(X)\mathbf{1}\{\mu_u(P) \in I\}\{u(P, Y) - \mu_u(P)\}]| \tag{4}$$

Here, $\mathbf{1}\{\mu_u(P) \in I\}$ equals 1 if $\mu_u(P) \in I$ and 0 otherwise. Small UC means that predicted mean utility agrees with realized utility in each predicted-utility interval. Hegazy et al. (2026) showed that the risk improvement achievable by monotone post-processing preserving the ordering of the one-dimensional score $\mu_u(P)$ is at most $2\,\mathrm{UC}(f; \{1\}, \{u\})$.

2.2 Cancellation in UC and Requirements for Hierarchical Auditing

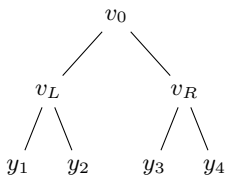
In applications with label trees, detecting only the final utility discrepancy is insufficient. For example, suppose the benefit of an intervention is overestimated among patients with the same predicted utility. Whether the error lies in an upper-level disease-group branch or in a subsequent severity branch determines which part of the model should be inspected and which probability should be corrected. We use a four-leaf tree to show that UC can lose this location information and explain why neither leaf-wise auditing nor direct auditing of all branches is sufficient to recover it.

Cancellation on a four-leaf tree. Consider the four-leaf labels in Figure 1, $\mathcal{Y} = \{y_1, y_2, y_3, y_4\}$, arranged in a binary tree. Condition on observations for which the predictor outputs the leaf-probability vector $P = p = (p_1, p_2, p_3, p_4)$, where $p_i > 0$. The root v_0 separates $\mathcal{Y}_L := \{y_1, y_2\}$ from $\mathcal{Y}_R := \{y_3, y_4\}$, and v_L and v_R separate the two leaves in their respective subtrees. For $t \in \{0, L, R\}$, denote the left and right children of v_t by $v_{t,L}$ and $v_{t,R}$. Thus $v_{0,L} = v_L$, $v_{0,R} = v_R$, and $(v_{L,L}, v_{L,R}, v_{R,L}, v_{R,R}) = (y_1, y_2, y_3, y_4)$.

For $j \in \{L, R\}$, let $q_{t,j}$ and $q^\star_{t,j}$ be the predicted and true probabilities of selecting $v_{t,j}$ conditional on reaching v_t, the latter within the group $P = p$. The left-branch probabilities satisfy $q_{t,L} = 1 - q_{t,R}$ and $q^\star_{t,L} = 1 - q^\star_{t,R}$. Assume $0 < q^\star_{0,R} < 1$, so both true lower-level branch probabilities exist.

Let $\mathcal{Y}_{v_{t,j}}$ be the leaves below child $v_{t,j}$ and assign it the mean utility

$$\bar{u}_{t,j} := \textstyle\sum_{y\in\mathcal{Y}_{v_{t,j}}} p_y u(p, y) / \sum_{y\in\mathcal{Y}_{v_{t,j}}} p_y, \qquad t \in \{0, L, R\}, \quad j \in \{L, R\}.$$



Right-branch probabilities q: predicted, $q^\star$: true

$$
\begin{aligned}
v_0: &\quad q_{0,R} = p_3 + p_4, &\quad q^\star_{0,R} &= \mathbb{P}(Y \in \mathcal{Y}_R \mid P = p),\\
v_L: &\quad q_{L,R} = \frac{p_2}{p_1 + p_2}, &\quad q^\star_{L,R} &= \mathbb{P}(Y = y_2 \mid Y \in \mathcal{Y}_L, P = p),\\
v_R: &\quad q_{R,R} = \frac{p_4}{p_3 + p_4}, &\quad q^\star_{R,R} &= \mathbb{P}(Y = y_4 \mid Y \in \mathcal{Y}_R, P = p).
\end{aligned}
$$

Figure 1: A four-leaf tree with predicted and true conditional branch probabilities at each node.

This marginalizes the target utility within the child subtree under the predictive conditional distribution; for a leaf child y_i, it equals $u_i := u(p, y_i)$. We suppress dependence on the fixed u, p. The predicted mean utility at the root is $\mu_u(p) = \sum_{j=L,R} q_{0,j}\bar{u}_{0,j}$.

The UC residual $u(p, y) - \mu_u(p)$ is the difference between the realized utility at the leaf and the utility assessed at the root, before the label's location is refined. Its conditional mean $D^{\text{full}}(p) := \mathbb{E}[u(p, Y) \mid P = p] - \mu_u(p)$ decomposes as $D^{\text{full}}(p) = \sum_{t\in\{\text{top},L,R\}} D_t(p)$,

$$
D_{\text{top}}(p) := \sum_{j=L,R} q^\star_{0,j}\bar{u}_{0,j} - \sum_{j=L,R} q_{0,j}\bar{u}_{0,j}, \quad D_t(p) := q^\star_{0,t}\Big[\sum_{j=L,R} q^\star_{t,j}\bar{u}_{t,j} - \sum_{j=L,R} q_{t,j}\bar{u}_{t,j}\Big], \tag{5}
$$

where $t \in \{L, R\}$. Each term compares the same marginalized child utility values $\bar{u}_{t,j}$ averaged under the true and predicted branch probabilities, weighted by the true probability of reaching the parent: 1 at the root and $q^\star_{0,L}, q^\star_{0,R}$ at the lower nodes. Equivalently, if identifying child $v_{t,j}$ reveals utility $\bar{u}_{t,j}$, each term measures whether the parent-level utility assessment agrees on average with the assessment after this refinement. Thus, the root-to-leaf mean discrepancy splits into branching-step mean discrepancies; see Appendix B.4 for the proof.

For a general predictor, let $D^{\text{full}}(P) := \mathbb{E}[u(P, Y) - \mu_u(P) \mid P]$. Because the interval indicator is determined by P alone, the tower property and Eq. (5) imply

$$
\text{UC}(f; \{1\}, \{u\}) = \sup_{I\in\mathcal{I}} \big|\mathbb{E}[\mathbf{1}\{\mu_u(P) \in I\} D^{\text{full}}(P)]\big| = \sup_{I\in\mathcal{I}} \big|\mathbb{E}[\mathbf{1}\{\mu_u(P) \in I\} \sum_{t\in\{\text{top},L,R\}} D_t(P)]\big|.
$$

Thus, UC evaluates the node contributions only after adding them within the same predicted-utility interval. If positive and negative contributions cancel (e.g., $D_{\text{top}} = 0.1$, $D_L = -0.1$, $D_R = 0$ gives $D^{\text{full}} = 0$), errors may remain at individual hierarchical levels even when UC is small. Moreover, even when UC is large, its value alone does not identify which internal decision should be inspected or corrected within the same target population.

From standard UC to branch auditing. [1] In medicine, the final utility guides decisions, while intermediate hierarchical stages help interpret decision paths and locate errors (Yu et al., 2025); in hierarchical selective classification, stage-wise confidence determines prediction granularity under uncertainty (Goren et al., 2024). This motivates auditing the local contributions underlying utility assessment, in addition to final-utility consistency.

Class-wise calibration of individual leaves does not directly guarantee the tree's conditional structure. Likewise, calibrating indicator utilities for nested subtrees with UC generally does not determine the branch probability conditional on reaching a parent node. One could instead add utilities designed to represent such conditional branch probabilities and calibrate each internal node using standard UC. However, what matters is not the branch-probability discrepancy itself, but its effect on the target utility. If the mean utilities assigned to a node's child subtrees are equal, an incorrect branch probability has no effect on their average. Thus, the node with the largest branch-probability discrepancy need not have the largest utility contribution or be the node that should be corrected.

Moreover, branch-specific predicted values define different audit populations, whereas we compare error sources within the common cell $\{\mu_u(P) \in I\}$. Auditing all internal nodes includes utility-irrelevant nodes and their finite-sample fluctuations, giving approximately $|\mathcal{U}||V^\circ|$ candidates, where V° is the internal-node set. Thus, we localize the utility-relevant contributions in Eq. (5) before cancellation, keeping the target utility and predicted-utility cell fixed.

[1] Detailed discussions of these points are provided in Appendices D.2–D.4.

3 HIERARCHICAL UTILITY CALIBRATION

Section 2.2 showed that UC can obscure which internal nodes generate a target-utility error. We now extend the four-leaf decomposition to a general tree and define HUC from node contributions evaluated with the same subgroup, target utility, and predicted-utility interval.

3.1 UTILITY-ERROR DECOMPOSITION AND HUC ON GENERAL TREES

Let $\mathcal{T} = (V, E_{\mathcal{T}})$ be a fixed finite rooted tree with leaf set $\mathcal{Y}$, root ρ, and internal-node set V°. For $a \in V$, let $\mathcal{Y}_a$ be the leaves in the subtree rooted at a, with $\mathcal{Y}_y = \{y\}$ for a leaf y. Each internal node v has children $v_1, \ldots, v_{d_v}$, whose leaf sets form a disjoint partition of $\mathcal{Y}_v$.

Assume $P = f(X)$ satisfies $P_y > 0$ almost surely for every $y \in \mathcal{Y}$. We first define local quantities at a fixed $p \in \Delta^{K-1}$ with strictly positive coordinates, then substitute $p = P$ to audit the predictor. For $a \in V$, $v \in V^\circ$, and $j = 1, \ldots, d_v$, set

$$\pi_a(p) := \sum_{y \in \mathcal{Y}_a} p_y, \qquad \mu_{u,a}(p) := \frac{\sum_{y \in \mathcal{Y}_a} p_y u(p,y)}{\pi_a(p)}, \qquad q_{v,j}(p) := \frac{\pi_{v_j}(p)}{\pi_v(p)}. \tag{6}$$

These are the predictive reach probability, subtree mean utility, and branch probability given the parent, respectively. Write $\boldsymbol{q}_v(p) := (q_{v,1}(p), \ldots, q_{v,d_v}(p))^\top$. Then $\sum_j q_{v,j}(p) = 1$, $\mu_{u,\rho}(p) = \mu_u(p)$, and $\mu_{u,y}(p) = u(p,y)$ at each leaf y.

The child utility values in Section 2.2 are $\bar{u}_{t,j} = \mu_{u,v_{t,j}}(p)$. For each $a \in V$, let $J_a(y) := \mathbf{1}\{y \in \mathcal{Y}_a\}$ indicate whether y lies in the subtree rooted at a. For an internal node v with child v_j, write $J_{v,j}(y) := J_{v_j}(y) = \mathbf{1}\{y \in \mathcal{Y}_{v_j}\}$, and define the branch residual and utility contrast by

$$\boldsymbol{R}_v(p,y) := \big(J_{v,j}(y) - J_v(y) q_{v,j}(p)\big)_{j=1}^{d_v}, \qquad \boldsymbol{\Delta}_{u,v}(p) := \big(\mu_{u,v_j}(p) - \mu_{u,v}(p)\big)_{j=1}^{d_v}. \tag{7}$$

For $a \in V$, let $\pi_a^\star(p) := \mathbb{P}(Y \in \mathcal{Y}_a \mid P = p)$ be the true reach probability. If $\pi_v^\star(p) > 0$, define $q_{v,j}^\star(p) := \pi_{v_j}^\star(p)/\pi_v^\star(p)$. The conditional mean of the node contribution satisfies

$$\mathbb{E}[\langle \boldsymbol{\Delta}_{u,v}(p), \boldsymbol{R}_v(p,Y)\rangle \mid P = p] = \pi_v^\star(p)\Big[\sum_{j=1}^{d_v} q_{v,j}^\star(p)\mu_{u,v_j}(p) - \sum_{j=1}^{d_v} q_{v,j}(p)\mu_{u,v_j}(p)\Big]. \tag{8}$$

If $\pi_v^\star(p) = 0$, the conditional mean is zero without defining $q_{v,j}^\star(p)$. See Lemma C.4 in Appendix C.2 for the proof. Each child utility value $\mu_{u,v_j}(p)$ marginalizes $u(p,y)$ within $\mathcal{Y}_{v_j}$ using the predictive conditional probabilities $p_y/\pi_{v_j}(p)$. The two sums in brackets of Eq. (8) average these same values under the true and predicted branch probabilities. Their difference measures mean-utility disagreement at this branching step, weighted by the true reach probability $\pi_v^\star(p)$. These conditional means generalize $D_{\mathrm{top}}(p)$, $D_L(p)$, and $D_R(p)$ in Eq. (5). The same inner products decompose the utility residual pointwise.

Theorem 3.1 (Utility-error decomposition on a general tree)**.** *For every target utility u, every $p \in \Delta^{K-1}$ with strictly positive coordinates, and every $y \in \mathcal{Y}$,*

$$u(p,y) - \mu_u(p) = \sum_{v \in V^\circ} \langle \boldsymbol{\Delta}_{u,v}(p), \boldsymbol{R}_v(p,y)\rangle\,. \tag{9}$$

Each contribution equals the realized child's mean utility minus the parent's, or zero outside the parent subtree. Along the root-to-leaf path, intermediate means cancel, leaving the realized leaf utility minus the predicted root mean. This recovers the original residual and locates its branching-step contributions without changing the target utility. See Theorem C.3 in Appendix C.2; Appendix B.3 gives the binary-tree correspondence.

For fixed $\mathcal{T}$ and p, the node contributions are mutually orthogonal under the predictive distribution $\widetilde{Y} \sim p$. They form the unique decomposition whose components vanish outside their respective subtrees, are constant on each child subtree, and have zero p-mean; see Appendix C.3. Thus, the tree, predictive distribution, and target utility determine the node-wise attribution. Under the true distribution, however, nonzero expected node contributions can still cancel.

We now substitute $(p, y) = (P, Y)$ into Eq. (9) to audit the input-dependent prediction $P = f(X)$. Using the same audit conditions (c, u, I) as UC, define the node-wise audit moment for $v \in V^\circ$ by

$$\Gamma_{c,u,I,v}(f) := \mathbb{E}[c(X)\mathbf{1}\{\mu_u(P) \in I\} \langle \boldsymbol{\Delta}_{u,v}(P), \boldsymbol{R}_v(P, Y)\rangle]. \tag{10}$$

Multiplying the pointwise decomposition by $c(X)\mathbf{1}\{\mu_u(P) \in I\}$ and taking expectations gives

$$\mathbb{E}[c(X)\mathbf{1}\{\mu_u(P) \in I\}\{u(P, Y) - \mu_u(P)\}] = \sum_{v \in V^\circ} \Gamma_{c,u,I,v}(f). \tag{11}$$

UC takes the absolute value after summing these moments. In contrast, the HUC error defined below audits branching-step changes in utility assessment separately under the same (c, u, I), preventing cancellation across nodes. The sum in Eq. (11) can be restricted to the utility-relevant nodes

$$\mathcal{V}_{\mathcal{T}}(u) := \{v \in V^\circ : \boldsymbol{\Delta}_{u,v} \not\equiv \mathbf{0}\}. \tag{12}$$

Here, $\boldsymbol{\Delta}_{u,v} \not\equiv \mathbf{0}$ means that the child means $\mu_{u,v_1}(p), \ldots, \mu_{u,v_{d_v}}(p)$ are not all equal for some strictly positive prediction p. Thus, $\mathcal{V}_{\mathcal{T}}(u)$ contains precisely the nodes where identifying the child containing the label can change the target-utility assessment; all other nodes contribute zero.

Definition 3.2 (Hierarchical Utility Calibration)**.** *The HUC error is*

$$\mathrm{HUC}(f; \mathcal{C}, \mathcal{U}) := \sup_{c \in \mathcal{C},\, u \in \mathcal{U},\, I \in \mathcal{I},\, v \in \mathcal{V}_{\mathcal{T}}(u)} |\Gamma_{c,u,I,v}(f)|. \tag{13}$$

Intuitively, the utility assessed at the parent should agree on average, within the audited population, with the utility assessed after learning which child contains the realized label. HUC evaluates this mean consistency node by node under the same (c, u, I). Partial sums control calibration errors for subtree-mean utilities at fixed boundaries of varying branch depths, with bounds proportional to the included utility-relevant node count; see Appendix G.

The following proposition establishes the relationships among UC, HUC, and canonical calibration.

Proposition 3.3. *For every subgroup class $\mathcal{C}$ and target utility u,*

$$\mathrm{UC}(f; \mathcal{C}, \{u\}) \le |\mathcal{V}_{\mathcal{T}}(u)|\ \mathrm{HUC}(f; \mathcal{C}, \{u\}). \tag{14}$$

Moreover, if f is canonically calibrated, then $\mathrm{HUC}(f; \{1\}, \mathcal{U}) = 0$.

See Appendix D.1 for the proof. Appendix G further relates HUC to monotone post-processing guarantees when the target utility is used only down to an internal-subtree boundary.

To estimate HUC, let $(X_i, Y_i)_{i=1}^n$ be independent and identically distributed (i.i.d.) copies of (X, Y), and write $P_i := f(X_i)$. Define

$$\widehat{\mathrm{HUC}}(f; \mathcal{C}, \mathcal{U}) := \sup_{c \in \mathcal{C},\, u \in \mathcal{U},\, I \in \mathcal{I}, v \in \mathcal{V}_{\mathcal{T}}(u)} \left| \frac{1}{n} \sum_{i=1}^{n} c(X_i)\mathbf{1}\{\mu_u(P_i) \in I\} \langle \boldsymbol{\Delta}_{u,v}(P_i), \boldsymbol{R}_v(P_i, Y_i)\rangle \right|.$$

Theorem 3.4 (Finite-sample evaluation of HUC)**.** *Assume that f, $\mathcal{T}$, and the finite classes $\mathcal{C}$,$\mathcal{U}$ are fixed independently of the audit sample. For any $\delta \in (0, 1)$, with probability at least $1 - \delta$,*

$$\left| \widehat{\mathrm{HUC}}(f; \mathcal{C}, \mathcal{U}) - \mathrm{HUC}(f; \mathcal{C}, \mathcal{U}) \right| \le \frac{16}{\sqrt{n}} + 4\sqrt{\frac{\log(\max(1, |\mathcal{C}| \sum_{u \in \mathcal{U}} |\mathcal{V}_{\mathcal{T}}(u)|)/\delta)}{2n}}. \tag{15}$$

The proof and a comparison of the computational costs of estimating UC and HUC are given in Appendices F.1 and F.4, respectively.

3.2 HUC-Boost: Branch-Local Post-Hoc Correction

Given a base predictor, we propose HUC-Boost, a boosting-style post-hoc correction algorithm for reducing its HUC error; Algorithm 1 gives the population version. As in the UC post-processing algorithm of Hegazy et al. (2026) (hereafter UC-Boost), each iteration selects an audit candidate whose moment exceeds a prespecified tolerance in magnitude and updates the predictor in the corresponding direction. Unlike UC-Boost, HUC-Boost changes branch probabilities only at the internal node with the selected violation.

To implement this node-local update, we parameterize the predictor by node-wise logits. At iteration t, let $\boldsymbol{z}_{t,v}(x) \in \mathbb{R}^{d_v}$ be the logits at node v, with corresponding branch probabilities $q_{t,v,j}(x) := e^{z_{t,v,j}(x)} / \sum_{k=1}^{d_v} e^{z_{t,v,k}(x)}$ for $j = 1, \ldots, d_v$. For each leaf y, the prediction $f_t(x)_y$ is the product of these branch probabilities along the path from the root to y. Appendix E.1 gives this construction and verifies its consistency with the branch probabilities.

For $a = (c, u, I, v)$, the node-v logit direction used in Step 4 of Algorithm 1 is

$$\boldsymbol{h}_a^t(x) := c(x)\mathbf{1}\{\mu_u(f_t(x)) \in I\}\boldsymbol{\Delta}_{u,v}(f_t(x)).$$

Algorithm 1 HUC-Boost: logit update (population version)

Input: Positive initial predictor f_0, candidate classes $\mathcal{C}, \mathcal{U}, \mathcal{I}$, tolerance $\varepsilon > 0$

1: $z_{0,v,j}(x) \leftarrow \log q_{v,j}(f_0(x))$ for all v, j; $t \leftarrow 0$
2: **while** there exists a candidate with $|\Gamma_{c,u,I,v}(f_t)| > \varepsilon$ **do**
3: Select any such $a_t = (c_t, u_t, I_t, v_t)$
4: $\boldsymbol{h}_t \leftarrow \boldsymbol{h}_{a_t}^t$
5: $\Gamma_t := \Gamma_{c_t,u_t,I_t,v_t}(f_t),\ \Lambda_t := \frac{1}{4}\mathbb{E}\left[J_{v_t}(Y)\{\max_j h_{t,j}(X) - \min_j h_{t,j}(X)\}^2\right],\ \alpha_t := \Gamma_t/\Lambda_t$
6: $\boldsymbol{z}_{t+1,v_t}(x) \leftarrow \boldsymbol{z}_{t,v_t}(x) + \alpha_t \boldsymbol{h}_t(x)$ for all x; keep all other nodes fixed
7: Reconstruct f_{t+1} from the softmax branch probabilities by path products; $t \leftarrow t + 1$
8: **end while**
9: **return** f_t

For the candidate a_t selected in Step 3, write $v := v_t$. The logit update in Step 6 and reconstruction in Step 7 give, for $y \in \mathcal{Y}_{v_j}$,

$$q_{t+1,v,j}(x) = \frac{q_{t,v,j}(x)e^{\alpha_t h_{t,j}(x)}}{\sum_k q_{t,v,k}(x)e^{\alpha_t h_{t,k}(x)}}, \qquad f_{t+1}(x)_y = f_t(x)_y \frac{q_{t+1,v,j}(x)}{q_{t,v,j}(x)}.$$

The update redistributes the predicted mass reaching v among its child subtrees. It rescales all leaf probabilities within each child subtree by the same factor, preserving their conditional distribution, the reach probability of v, and all leaf probabilities outside $\mathcal{Y}_v$.

The step size in Algorithm 1 is justified by the global leaf log loss: Γ_t is its negative directional derivative along the fixed direction $\boldsymbol{h}_t$, and $\Lambda_t > 0$ bounds the curvature whenever an update is made. This yields the following guarantee.

Theorem 3.5 (Loss decrease and finite termination)**.** *Let $\mathcal{L}_{\log}(f) := \mathbb{E}[-\log f(X)_Y]$. Assume $\mathcal{L}_{\log}(f_0) < \infty$. Algorithm 1 decreases population log loss monotonically. It terminates within $T \le \lceil 2\mathcal{L}_{\log}(f_0)/\varepsilon^2 \rceil$ updates and returns f_T satisfying $\mathrm{HUC}(f_T; \mathcal{C}, \mathcal{U}) \le \varepsilon$.*

Appendix E.2 gives the derivations and proof of Theorem 3.5. The auditing implementation and theoretical analysis of finite-sample HUC-Boost are given in Appendices F.3 and H, respectively. Appendix E.3 discusses comparisons with UC-Boost and Brier-based alternatives.

4 Related Work

Prior work covers confidence and class-wise calibration (Guo et al., 2017; Vaicenavicius et al., 2019), multicalibration within prespecified subgroups (Hebert-Johnson et al., 2018), and kernel-based calibration tests (Widmann et al., 2019). Canonical calibration conditions on the entire predictive vector, but its distribution-free testing can require exponentially many samples in the number of classes (Gopalan et al., 2024). Utility Calibration (UC) audits prespecified utilities over predicted-utility intervals (Hegazy et al., 2026), generalizing binary Cutoff Calibration Error (Rossellini et al., 2025). Other decision-oriented approaches include decision calibration for specified losses (Zhao et al., 2021), task-specific kernel calibration metrics (Marx et al., 2023), and decision calibration for nonlinear losses (Tang et al., 2026). These works study which decisions and losses predictions can reliably support and which calibration conditions are needed. Our focus is complementary: we decompose the residual of the same specified target utility along a semantic label hierarchy and

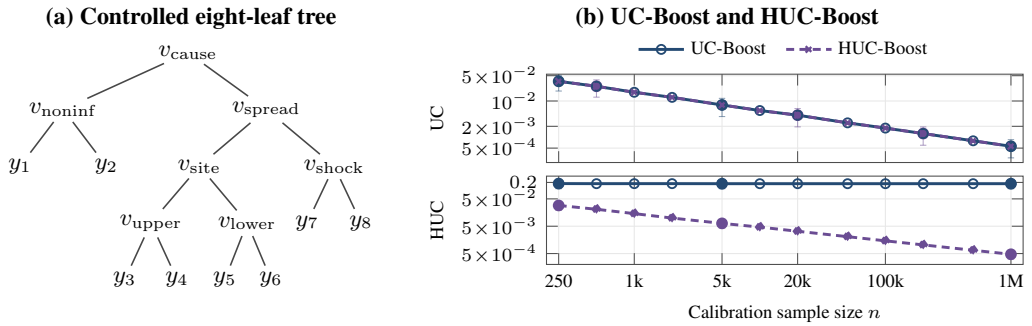


Figure 2: Toy tree and population UC/HUC after UC- and HUC-Boost with n calibration labels.

evaluate its node contributions before cancellation, under the same subgroup and predicted-utility interval. Additional comparisons with related work appear in Appendix I.

Hierarchical classification has a long history, including methods that place classifiers at the nodes of a hierarchy, share statistical information between parents and children, or learn a structured classifier over the hierarchy (Koller & Sahami, 1997; McCallum et al., 1998; Cesa-Bianchi et al., 2006; Rousu et al., 2006; Silla & Freitas, 2011). Further work has studied classification consistent with a hierarchical distance (Ramaswamy et al., 2015), methods that return an ancestor when the prediction is uncertain in order to trade specificity for accuracy (Deng et al., 2012; Goren et al., 2024), set-valued predictions represented by internal nodes or multiple nodes (Mortier et al., 2022), and decoding rules adapted to hierarchical evaluation measures (Plaud et al., 2025). Most of these methods use the hierarchy in classifier training, loss design, or output selection, rather than addressing probabilistic calibration. In particular, the calibrated surrogates of Ramaswamy et al. (2015). refer to consistency with respect to a hierarchical classification loss and are different from the probabilistic calibration studied here. Work that directly addresses predictive-probability calibration on a hierarchy includes methods that calibrate the internal binary classifiers of a nested dichotomy or the probability distribution produced by the complete tree (Leathart et al., 2019). In contrast, HUC localizes utility errors to relevant nodes without requiring calibration of every internal branch probability.

5 Experiments

5.1 Toy data

Setup. In a controlled toy data setting, we test whether the cancellation between internal-node contributions discussed in Sections 2.2 and 3.1 allows UC-Boost to reduce UC while leaving nonzero HUC. We use the medically motivated synthetic eight-leaf tree in Figure 2(a), with seven internal nodes, three decision utilities, and whole-population audits. Each utility evaluates a binary action choice based on outcomes distinguished by the tree. For each sample size n, UC- and HUC-Boost share the initial prediction and the n calibration labels. Figure 2(b) reports population UC (top) and HUC (bottom) under the known label distribution. Curves show means over 200 repetitions; error bars at selected sizes show one sample standard deviation. Appendix J.1 gives the full setup.

Results. The UC errors after UC- and HUC-Boost are similar and decrease as n grows. Their HUC errors differ: after UC-Boost, HUC remains, whereas after HUC-Boost it decreases with sample size. HUC-Boost reduces both errors by addressing node contributions before cancellation. This is consistent with Proposition 3.3, which bounds UC by a multiple of HUC. In contrast, UC-Boost reduces the sum without necessarily reducing each contribution, illustrating the decomposition in Eq. (11) and the cancellation discussed in Section 2.2.

5.2 Real data

Setup. We compare HUC-Boost with other post-hoc methods on real datasets. SUPPORT2 (Knaus et al., 1995) contains tabular data from seriously ill hospitalized adults. Five outcomes are arranged in a medically motivated binary tree, and nine utilities are obtained from three medical action com-

Table 1: Independent-test results: mean ± sample standard deviation over six base predictors and five splits. UC-Boost and HUC-Boost are abbreviated as UC-B and HUC-B, respectively.

	iNaturalist					SUPPORT2				
Method	Acc. ↑	AUC ↑	UC ↓	HUC ↓	Updates	Acc. ↑	AUC ↑	UC ↓	HUC ↓	Updates
Baseline	.489±.090	.900±.108	.272±.161	.125±.059	0.0±0.0	.499±.199	.676±.083	.167±.178	.148±.142	0.0±0.0
UC-B	.462±.088	.914±.066	**.045±.018**	.036±.011	25.4±24.1	.575±.051	.681±.079	.039±.014	.050±.031	33.9±39.4
HUC-B	.463±.090	.922±.060	.075±.032	.034±.009	145.9±180.5	.575±.045	.698±.060	.036±.010	.035±.009	45.2±46.9
Temp.	.489±.090	.890±.110	.130±.114	.092±.076	0.0±0.0	.499±.199	.686±.072	.077±.069	.076±.069	0.0±0.0
Temp.→HUC	.470±.088	**.930±.050**	.047±.014	**.029±.008**	43.1±25.9	.587±.032	.703±.057	.032±.006	.032±.006	20.4±22.6
Vec.	**.489±.095**	.899±.100	.114±.092	.083±.073	0.0±0.0	.575±.053	.691±.065	.038±.015	.038±.015	0.0±0.0
Vec.→HUC	.478±.095	.929±.053	.050±.018	.031±.008	28.2±22.4	.589±.032	.701±.051	**.031±.008**	**.031±.008**	14.3±12.5
Dir.	.436±.112	.912±.063	.105±.053	.055±.023	0.0±0.0	.579±.046	.712±.066	.035±.013	.035±.013	0.0±0.0
Dir.→HUC	.435±.112	.912±.062	.109±.052	.056±.023	4.3±10.5	**.589±.032**	**.718±.050**	.032±.008	.032±.007	13.3±14.1

parisons evaluated at three parameter values. On iNaturalist (Van Horn et al., 2018), images are classified into 30 species using a nonuniform biological taxonomy of depth seven, and top-1 correctness at the class, order, family, genus, and species levels is used as the utility. Appendix K provides detailed experimental settings and experiments on additional real datasets.

Both datasets use six base predictors—logistic regression, Gaussian naive Bayes, decision tree, random forest, XGBoost, and multilayer perceptron—using five splits into training, calibration, validation, and test sets. For iNaturalist, each image is converted to a 512-dimensional feature vector using an ImageNet-pretrained ResNet-18, and the base predictors are trained on these features. Base predictors are fitted on the training data; UC-Boost, HUC-Boost, and other recalibration methods are fitted on the calibration data. Validation data select regularization and the checkpoint returned along each boosting path, and test data are used only for final evaluation.

We compare a base predictor (Baseline), UC- and HUC-Boost, temperature and vector scaling (Guo et al., 2017), and Dirichlet calibration (Kull et al., 2019) (see Appendix K for details). We also apply HUC-Boost after each parametric method, using the same calibration data for both stages; see Appendix H for the algorithm and its theoretical properties. The first-stage parametric recalibrator adjusts the predictive distribution globally, and HUC-Boost then targets the remaining node-wise utility errors locally. This global–local combination is expected to improve recalibration efficiency.

Results. Table 1 reports independent-test metrics and boosting update counts. Direct HUC-Boost substantially reduces HUC relative to Baseline in both datasets. It uses more updates than UC-Boost, consistent with their update scopes: UC-Boost adjusts all leaf logits globally, whereas HUC-Boost changes branch logits at one internal node per update.

All two-stage variants use substantially fewer HUC-Boost updates than direct HUC-Boost. HUC-Boost after temperature or vector scaling yields lower mean HUC and higher mean AUC in both datasets. The Dirichlet-based variant improves these metrics only on SUPPORT2.

6 Conclusion and Limitations

In this paper, we have shown that, in multiclass probabilistic prediction with hierarchical labels, the utility residual decomposes into internal-node contributions. Based on this decomposition, we proposed HUC to evaluate these contributions before cancellation and HUC-Boost to correct violated branches. We further presented a two-stage recalibration algorithm that combines parametric recalibration methods with HUC-Boost. Numerical experiments demonstrated the effectiveness of the proposed approach in reducing hierarchical utility error.

Our analysis assumes a semantically fixed finite rooted label tree and a utility class specified before analysis. It therefore does not cover decisions outside $\mathcal{U}$, relationships absent from the tree, or settings in which the hierarchical structure changes dynamically. Moreover, as with Utility Calibration, our method concerns the agreement between predicted and realized utilities in expectation and does not by itself guarantee the optimality of the resulting decisions. Addressing these limitations will be an important direction for future work toward more reliable decision-making.

AI USE STATEMENT

In this work, we used generative AI tools for generating synthetic datasets, helping develop theoretical models or conceptual frameworks, assisting in the writing of proofs, designing or providing feedback on research methodology or experiments, implementing methods, assisting with translation, and cleaning and reformatting datasets. We have not used generative AI tools for formulating mathematical claims, providing critical ingredients for proving mathematical claims, proposing or refining hypotheses, supporting qualitative and thematic data analysis, or interpreting results, and all tasks with required disclosure are applicable to this work. Additionally, we used generative AI tools for creating or modifying scientific figures or images, suggesting experimental parameters, creating or editing software code, creating artifacts, summarizing or analysing existing literature, sourcing or searching for information, editing the paper to improve readability, and formatting references. We have reviewed all AI-assisted work. Specifically, we checked all mathematical arguments, verified each reference individually against its original source, and tested the implementation used for the reported experiments. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work develops methods for recalibrating probabilistic predictions and introduces no new human-subject data collection. Existing benchmarks, including SUPPORT2, are used for methodological evaluation under their published access conditions. The medically motivated label hierarchies and action utilities in the toy and SUPPORT2 experiments are synthetic research constructions, not treatment effects or clinical recommendations. HUC evaluates calibration with respect to a specified label tree, utility class, and subgroup class; a small HUC error does not by itself establish clinical effectiveness, fairness, or decision optimality.

REPRODUCIBILITY STATEMENT

The definitions, assumptions, and proofs underlying our main theoretical results are provided in Appendices A–H. Appendix J specify the toy and additional experiments, while Appendix K describe the real-data benchmarks, preprocessing, label hierarchies, utilities, subgroups, data splits, recalibration protocols, and additional by-base results.

REFERENCES

Emanuele Bastianelli, Andrea Vanzo, Pawel Swietojanski, and Verena Rieser. SLURP: A spoken language understanding resource package. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 7252–7262, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.588. URL https://aclanthology.org/2020.emnlp-main.588.

Nicoló Cesa-Bianchi, Claudio Gentile, and Luca Zaniboni. Incremental Algorithms for Hierarchical Classification. *Journal of Machine Learning Research*, 7(2):31–54, 2006. URL http://jmlr.org/papers/v7/cesa-bianchi06a.html.

Santiago Cortes-Gomez, Carlos Patiño, Yewon Byun, Steven Wu, Eric Horvitz, and Bryan Wilder. Utility-Directed Conformal Prediction: A Decision-Aware Framework for Actionable Uncertainty Quantification. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 4227–4246, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/0c6b452f1bbfb6905f6bac957d73b321-Paper-Conference.pdf.

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*, pp. 248–255, 2009. doi: 10.1109/CVPR.2009.5206848. URL https://ieeexplore.ieee.org/document/5206848.

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, pp. 3450–3457, 2012. doi: 10.1109/CVPR.2012.6248086.

Jack FitzGerald, Christopher Hench, Charith Peris, Scott Mackie, Kay Rottmann, Ana Sanchez, Aaron Nash, Liam Urbach, Vishesh Kakarala, Richa Singh, Swetha Ranganath, Laurie Crist, Misha Britan, Wouter Leeuwis, Gokhan Tur, and Prem Natarajan. MASSIVE: A 1M-Example Multilingual Natural Language Understanding Dataset with 51 Typologically-Diverse Languages. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 4277–4302, Toronto, Canada, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.235. URL https://aclanthology.org/2023.acl-long.235/.

Masahiro Fujisawa and Futoshi Futami. PAC-Bayes Analysis for Recalibration in Classification. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 17986–18023. PMLR, 2025. URL https://proceedings.mlr.press/v267/fujisawa25a.html.

Parikshit Gopalan, Lunjia Hu, Michael P. Kim, Omer Reingold, and Udi Wieder. Loss Minimization Through the Lens Of Outcome Indistinguishability. In Yael Tauman Kalai (ed.), *14th Innovations in Theoretical Computer Science Conference (ITCS 2023)*, volume 251 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pp. 60:1–60:20, Dagstuhl, Germany, 2023. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. ISBN 978-3-95977-263-1. doi: 10.4230/LIPIcs.ITCS.2023.60. URL https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.ITCS.2023.60.

Parikshit Gopalan, Lunjia Hu, and Guy N. Rothblum. On Computationally Efficient Multi-Class Calibration. In Shipra Agrawal and Aaron Roth (eds.), *Proceedings of Thirty Seventh Conference on Learning Theory*, volume 247 of *Proceedings of Machine Learning Research*, pp. 1983–2026. PMLR, 2024. URL https://proceedings.mlr.press/v247/gopalan24a.html.

Shani Goren, Ido Galil, and Ran El-Yaniv. Hierarchical Selective Classification. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 111047–111073. Curran Associates, Inc., 2024. doi: 10.52202/079017-3526. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/c8b100b376a7b338c84801b699935098-Paper-Conference.pdf.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On Calibration of Modern Neural Networks. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1321–1330. PMLR, 2017. URL https://proceedings.mlr.press/v70/guo17a.html.

Ursula Hebert-Johnson, Michael Kim, Omer Reingold, and Guy Rothblum. Multicalibration: Calibration for the (Computationally-Identifiable) Masses. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1939–1948. PMLR, 2018. URL https://proceedings.mlr.press/v80/hebert-johnson18a.html.

Mahmoud Hegazy, Michael I. Jordan, and Aymeric Dieuleveut. Scalable Utility-Aware Multiclass Calibration. In *Proceedings of The 29th International Conference on Artificial Intelligence and Statistics*, pp. 397–405. PMLR, 2026. URL https://proceedings.mlr.press/v300/hegazy26a.html.

Bobby Kleinberg, Renato Paes Leme, Jon Schneider, and Yifeng Teng. U-calibration: Forecasting for an unknown agent. In Gergely Neu and Lorenzo Rosasco (eds.), *Proceedings of Thirty Sixth Conference on Learning Theory*, volume 195 of *Proceedings of Machine Learning Research*, pp. 5143–5145. PMLR, 12–15 Jul 2023.

W. A. Knaus, F. E. Harrell, J. Lynn, L. Goldman, R. S. Phillips, A. F. Connors, N. V. Dawson, W. J. Fulkerson, R. M. Califf, N. Desbiens, P. Layde, R. K. Oye, P. E. Bellamy, R. B. Hakim, and D. P. Wagner. The SUPPORT prognostic model. Objective estimates of survival for seriously

ill hospitalized adults. Study to understand prognoses and preferences for outcomes and risks of treatments. *Annals of Internal Medicine*, 122(3):191–203, 1995. ISSN 0003-4819. doi: 10.7326/0003-4819-122-3-199502010-00007.

Daphne Koller and Mehran Sahami. Hierarchically Classifying Documents Using Very Few Words. In *International Conference on Machine Learning*, 1997. URL https://api.semanticscholar.org/CorpusID:2112467.

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. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/hash/8ca01ea920679a0fe3728441494041b9-Abstract.html.

Tim Leathart, Eibe Frank, Bernhard Pfahringer, and Geoffrey Holmes. On Calibration of Nested Dichotomies. In Qiang Yang, Zhi-Hua Zhou, Zhiguo Gong, Min-Ling Zhang, and Sheng-Jun Huang (eds.), *Advances in Knowledge Discovery and Data Mining*, pp. 69–80, Cham, 2019. Springer International Publishing. ISBN 978-3-030-16148-4. doi: 10.1007/978-3-030-16148-4_6.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to information retrieval*. Cambridge University Press, 2008. ISBN 978-0-521-86571-5. doi: 10.1017/CBO9780511809071. URL https://nlp.stanford.edu/IR-book/pdf/irbookprint.pdf.

Charlie Marx, Sofian Zalouk, and Stefano Ermon. Calibration by Distribution Matching: Trainable Kernel Calibration Metrics. In *Advances in Neural Information Processing Systems*, volume 36, pp. 25910–25928. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/52493d82db00e73abb2858a5a5f28717-Abstract-Conference.html.

Andrew McCallum, Ronald Rosenfeld, Tom M. Mitchell, and Andrew Y. Ng. Improving Text Classification by Shrinkage in a Hierarchy of Classes. In *Proceedings of the Fifteenth International Conference on Machine Learning*, ICML '98, pp. 359–367, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc. ISBN 1-55860-556-8.

Yair Meidan, Michael Bohadana, Yael Mathov, Yisroel Mirsky, Asaf Shabtai, Dominik Breitenbacher, and Yuval Elovici. N-BaIoT—Network-Based Detection of IoT Botnet Attacks Using Deep Autoencoders. *IEEE Pervasive Computing*, 17(3):12–22, 2018. ISSN 1558-2590. doi: 10.1109/MPRV.2018.03367731. URL https://ieeexplore.ieee.org/document/8490192.

Thomas Mortier, Eyke Hüllermeier, Krzysztof Dembczyński, and Willem Waegeman. Set-valued prediction in hierarchical classification with constrained representation complexity. In *Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence*, pp. 1392–1401. PMLR, 2022. URL https://proceedings.mlr.press/v180/mortier22a.html.

Roman Plaud, Alexandre Perez-Lebel, Matthieu Labeau, Antoine Saillenfest, and Thomas Bonald. To Each Metric Its Decoding: Post-Hoc Optimal Decision Rules of Probabilistic Hierarchical Classifiers. In *Proceedings of the 42nd International Conference on Machine Learning*, pp. 49493–49536. PMLR, 2025. URL https://proceedings.mlr.press/v267/plaud25a.html.

Harish Ramaswamy, Ambuj Tewari, and Shivani Agarwal. Convex Calibrated Surrogates for Hierarchical Classification. In Francis Bach and David Blei (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 1852–1860, Lille, France, 2015. PMLR. URL https://proceedings.mlr.press/v37/ramaswamy15.html.

Raphael Rossellini, Jake A. Soloff, Rina Foygel Barber, Zhimei Ren, and Rebecca Willett. Can a calibration metric be both testable and actionable? In *Proceedings of Thirty Eighth Conference on Learning Theory*, pp. 4937–4972. PMLR, 2025. URL https://proceedings.mlr.press/v291/rossellini25a.html.

Juho Rousu, Craig Saunders, Sandor Szedmak, and John Shawe-Taylor. Kernel-Based Learning of Hierarchical Multilabel Classification Models. *Journal of Machine Learning Research*, 7(59): 1601–1626, 2006. URL http://jmlr.org/papers/v7/rousu06a.html.

Carlos N. Silla and Alex A. Freitas. A survey of hierarchical classification across different application domains. *Data Mining and Knowledge Discovery*, 22(1):31–72, 2011. ISSN 1573-756X. doi: 10.1007/s10618-010-0175-9. URL https://doi.org/10.1007/s10618-010-0175-9.

Jingwu Tang, Jiayun Wu, Steven Wu, and Jiahao Zhang. Dimension-Free Decision Calibration for Nonlinear Loss Functions. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 86293–86321, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/8ad060589ed7e18aea4ee1cc8a486c89-Paper-Conference.pdf.

Mahbod Tavallaee, Ebrahim Bagheri, Wei Lu, and Ali A. Ghorbani. A detailed analysis of the KDD CUP 99 data set. In *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, CISDA 2009, Ottawa, Canada, July 8-10, 2009*, pp. 1–6. IEEE, 2009. doi: 10.1109/CISDA.2009.5356528. URL https://doi.org/10.1109/CISDA.2009.5356528.

Juozas Vaicenavicius, David Widmann, Carl Andersson, Fredrik Lindsten, Jacob Roll, and Thomas Schön. Evaluating model calibration in classification. In Kamalika Chaudhuri and Masashi Sugiyama (eds.), *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pp. 3459–3467. PMLR, 2019. URL https://proceedings.mlr.press/v89/vaicenavicius19a.html.

Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The INaturalist Species Classification and Detection Dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8769–8778, 2018. URL https://openaccess.thecvf.com/content_cvpr_2018/html/Van_Horn_The_INaturalist_Species_CVPR_2018_paper.html.

David Widmann, Fredrik Lindsten, and Dave Zachariah. Calibration tests in multi-class classification: A unifying framework. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d' Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/1c336b8080f82bcc2cd2499b4c57261d-Paper.pdf.

Zhen Yu, Toan D. Nguyen, Lie Ju, Yaniv Gal, Maithili Sashindranath, Paul Bonnington, Lei Zhang, Victoria Mar, and Zongyuan Ge. Hierarchical skin lesion image classification with prototypical decision tree. *npj Digital Medicine*, 8(1):26, 2025. ISSN 2398-6352. doi: 10.1038/s41746-024-01395-z. URL https://doi.org/10.1038/s41746-024-01395-z.

Shengjia Zhao, Michael Kim, Roshni Sahoo, Tengyu Ma, and Stefano Ermon. Calibrating Predictions to Decisions: A Novel Approach to Multi-Class Calibration. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 22313–22324. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/bbc92a647199b832ec90d7cf57074e9e-Paper.pdf.

GUIDE TO THE APPENDICES

Appendix	Content and proof locations
A: Formal definitions and tree properties	Defines the probability space, label-tree quantities, subtree mean utilities, and the path-product representation.
B: Binary classification and trees	Develops the binary-tree calculations and path decomposition. Appendix B.4 directly proves Eq. (5) and the UC identity in Section 2.2.
C: General multiway trees	Appendix C.1 proves Eq. (7); Appendix C.2 proves Theorem 3.1, Eqs. (9) and (8), and the mean-consistency decomposition; Appendix C.3 proves orthogonality, uniqueness, and the Parseval identity.
D: Basic properties of HUC	Appendix D.1 proves Proposition 3.3, Eq. (14), and the implication from canonical calibration. Appendices D.2–D.4 give counterexamples concerning conditional branches, utility impact, and node-specific audit populations.
E: Proof of HUC-Boost	Appendices E.1 and E.2 analyze Algorithm 1 and prove Theorem 3.5; Appendix E.3 gives the ordinary-leaf-Brier comparison; Appendix E.4 proves the path-product results.
F: Finite-sample evaluation	Appendix F.1 proves Theorem 3.4; Appendix F.3 derives the exact tie-preserving scan; Appendix F.4 derives the exact UC–HUC operation counts.
G: Post-processing and HUC	Uses the decomposition in Appendix C.2 to prove calibration and one-step refinement guarantees for subtree-mean utilities, including Proposition G.3, and relates them to monotone post-processing.
H: Empirical HUC-Boost and fixed-step sample complexity	Algorithm 3 gives the adaptive- and fixed-step implementations. Appendices H.1–H.4 prove fixed-step sample-complexity results for fixed and learned initializers, including Algorithm 4.
I: Additional related work	Compares HUC with U-calibration, loss outcome indistinguishability and omniprediction, and utility-directed conformal prediction.
J: Toy data	Specifies the main eight-leaf correction experiment and the decision-utility construction.
K: Real data	Gives the common protocol and the data, hierarchies, utilities, subgroups, and results for iNaturalist, SUPPORT2, MASSIVE, NSL-KDD, and N-BaIoT.

A FORMAL DEFINITIONS AND BASIC PROPERTIES OF THE TREE

For a node v, let $\mathcal{Y}_v$ denote its descendant-leaf set. We use the subtree-membership indicators $J_v(y) := \mathbf{1}\{y \in \mathcal{Y}_v\}$, $J_{v,j}(y) := J_{v_j}(y) = \mathbf{1}\{y \in \mathcal{Y}_{v_j}\}$, where v_j is a child of v. Each indicator is one when y belongs to the corresponding subtree and zero otherwise.

This notation is used throughout the remaining appendices unless stated otherwise.

A.1 UTILITY FUNCTIONS AND DECISION-MAKING

Utility and its forecast. In UC, a utility evaluates the consequence of using p when y is observed, possibly incorporating a decision (Hegazy et al., 2026, Sections 3 and 3.3):

$$u : \Delta^{K-1} \times \mathcal{Y} \to [-1, 1], \qquad \mu_u(p) = \mathbb{E}_{\widetilde{Y}\sim p}[u(p, \widetilde{Y})] = \sum\nolimits_y p_y u(p, y).$$

Its utility classes are built from outcome values, ranks, or action payoffs and selection rules.

Linear utilities. A fixed payoff vector gives $u_b(p, y) = b_y$, $\mu_{u_b}(p) = \langle p, b \rangle$, $b \in [-1, 1]^K$. Varying b gives $\mathcal{U}_{\text{lin}}$; class-wise indicators use $b = e_j$. For a tree node v, let $\mathcal{Y}_v$ be its descendant-leaf set and write $J_v(y) := \mathbf{1}\{y \in \mathcal{Y}_v\}$ and $\pi_v(p) := \sum_{z \in \mathcal{Y}_v} p_z$. Thus J_v is a 0–1 indicator of subtree membership, with $\mathcal{Y}_y = \{y\}$ at a leaf. Subtree indicators $b_y = J_v(y)$ are instances.

Rank-based utilities. Let $r_p(y)$ be the rank of y under decreasing predicted probabilities. Then $u_\theta(p, y) = \theta_{r_p(y)}$, $\theta \in [-1, 1]^K$. Varying θ gives $\mathcal{U}_{\text{rank}}$. The choice $\theta_r = \mathbf{1}\{r \le k\}$ gives top-k correctness ($1 \le k \le K$), with top-class correctness at $k = 1$.

Decision-induced utilities. For a finite action set $\mathcal{A}$, a payoff $U : \mathcal{A} \times \mathcal{Y} \to [-1, 1]$ evaluates actions, whereas a measurable rule $\psi : \Delta^{K-1} \to \mathcal{A}$ selects them. Set

$$u_\psi(p, y) = U(\psi(p), y), \qquad \psi_U(p) = \operatorname*{arg\,max}_{\prec;\, a \in \mathcal{A}} \sum\nolimits_z p_z U(a, z). \tag{16}$$

Varying U with $m = |\mathcal{A}|$ and $\psi = \psi_U$ gives UC's $\mathcal{U}_{\text{dec},m}$ (equivalently, $U(a, y) = -\ell(y, a)$). General u_ψ also permits other rules. *Output correctness* for a reported node v uses $U(v, y) = J_v(y) = \mathbf{1}\{y \in \mathcal{Y}_v\}$, hence

$$u_\psi(p, y) = J_{\psi(p)}(y) = \mathbf{1}\{y \in \mathcal{Y}_{\psi(p)}\}, \qquad \mu_{u_\psi}(p) = \sum_{z \in \mathcal{Y}_{\psi(p)}} p_z = \pi_{\psi(p)}(p).$$

The utility is one when the reported subtree contains the true label and zero otherwise; its predicted mean is the probability assigned to that subtree. This covers iNaturalist's taxonomic outputs (Appendix K.2) and the status, family, and leaf outputs of NSL-KDD and N-BaIoT (Appendices K.5 and K.6), using their respective reporting rules. *Binary action comparisons* in Toy and SUPPORT2 (Appendices J.1 and K.3) use ψ_U with synthetic payoffs

$$U(a, y) = (2a - 1)\{\mathbf{1}_S(y) - \tau \mathbf{1}_D(y)\}, \quad a \in \{0, 1\}, \quad S \subseteq D \subseteq \mathcal{Y}, \quad \tau \in [0, 1].$$

Abstention and specificity in MASSIVE (Appendix K.4) use the prescribed confidence-threshold rule in Eq. (212) and graded payoffs in Eq. (213), rather than requiring $\psi = \psi_U$.

UC audits the accuracy of μ_u as a utility forecast, not decision optimality. Our experiments specify finite target classes $\mathcal{U}$. Fix these and total orders $\prec$ independently of the audit sample; use the first maximizer and ordered ranks at ties. Measurable scores then give measurable rules.

A.2 Probability Space, Label Tree, and Predictor

Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space, with input $X : \Omega \to \mathcal{X}$ and label $Y : \Omega \to \mathcal{Y}$. Let $\mathcal{Y} = \{1, \dots, K\}$. Write the predictor and its output as $f : \mathcal{X} \to \Delta^{K-1}$, $P := f(X)$. Throughout the theory, assume that $P_y > 0$ almost surely for every $y \in \mathcal{Y}$. Whenever a deterministic prediction vector p is used, likewise assume that $p_y > 0$ for every y.

Fix a version of the regular conditional distribution of Y given P, and write

$$\eta_y(p) := \mathbb{P}(Y = y \mid P = p), \qquad y \in \mathcal{Y} \tag{17}$$

All subsequent identities conditioned on $P = p$ are understood to hold for almost every p under the law of P.

Definition A.1 (Canonical calibration)**.** *A predictor f is canonically calibrated if, for every $y \in \mathcal{Y}$,*

$$\mathbb{P}(Y = y \mid P) = P_y \qquad \mathbb{P}\text{-a.s.} \tag{18}$$

In the notation of Eq. (17)*, this is equivalent to $\eta_y(P) = P_y$ almost surely.*

Let $\mathcal{T} = (V, E_{\mathcal{T}})$ be a finite rooted tree with root ρ, leaf set $\mathcal{Y}$, and internal-node set V°. Every non-root node has a unique parent. Write the children of an internal node v as

$$\text{ch}(v) = \{v_1, \dots, v_{d_v}\}, \qquad d_v \ge 1 \tag{19}$$

Here $J_a(y) = \mathbf{1}\{y \in \mathcal{Y}_a\}$ indicates membership in the descendant-leaf set $\mathcal{Y}_a$ of node a, and $J_{v,j} = J_{v_j}$ for a child v_j. The number of children d_v may vary across nodes, and one-child nodes are allowed. If $d_v = 1$, then $q_{v,1} = 1$, $J_{v,1} = J_v$, and $\mu_{u,v_1} = \mu_{u,v}$, so $\boldsymbol{R}_v = \boldsymbol{\Delta}_{u,v} = \mathbf{0}$. Such

a node contributes to neither the decomposition nor HUC, is never updated, and may be retained or contracted.

For each node $a \in V$, define its descendant leaf set by

$$\mathcal{Y}_a := \{y \in \mathcal{Y} : \text{the path from } \rho \text{ to } y \text{ passes through } a\} \tag{20}$$

Thus $\mathcal{Y}_\rho = \mathcal{Y}$, and $\mathcal{Y}_y = \{y\}$ for a leaf y. Since the child subtrees of v have disjoint leaf sets,

$$\mathcal{Y}_v = \dot{\bigcup}_{j=1}^{d_v} \mathcal{Y}_{v_j}. \tag{21}$$

Fix a leaf $y \in \mathcal{Y}$. Write the *ordered sequence* of nodes on the unique path from the root ρ to y as

$$\big(w_0(y), w_1(y), \ldots, w_{L_y}(y)\big) \tag{22}$$

where

$$w_0(y) = \rho, \qquad w_{L_y}(y) = y, \qquad w_{r+1}(y) \in \mathrm{ch}\big(w_r(y)\big) \quad (r = 0, \ldots, L_y - 1) \tag{23}$$

For each $r = 0, \ldots, L_y - 1$, let

$$w_{r+1}(y) = \big(w_r(y)\big)_{j_r(y)} \tag{24}$$

define the unique index $j_r(y) \in \{1, \ldots, d_{w_r(y)}\}$ of the child selected by the path at $w_r(y)$. Write the set of internal nodes on the path as

$$\mathrm{Path}(y) := \{w_0(y), \ldots, w_{L_y-1}(y)\} \tag{25}$$

If $v = w_r(y) \in \mathrm{Path}(y)$, also write $j_v(y) := j_r(y)$. When computing for a fixed y, we sometimes abbreviate $w_r := w_r(y)$ and $j_r := j_r(y)$ for readability.

A.3 Tree Probabilities and Subtree Mean Utilities

For a prediction vector $p \in \Delta^{K-1}$ and a node $a \in V$, define the *predicted reach probability* of a by

$$\pi_a(p) := \sum_{y \in \mathcal{Y}_a} p_y \tag{26}$$

All leaf probabilities are positive, so $\pi_a(p) > 0$ for every node a. Moreover, $\mathcal{Y}_\rho = \mathcal{Y}$, $\mathcal{Y}_y = \{y\}$, and $p \in \Delta^{K-1}$ imply

$$\pi_\rho(p) = \sum_{z \in \mathcal{Y}_\rho} p_z = \sum_{z \in \mathcal{Y}} p_z = 1, \qquad \pi_y(p) = \sum_{z \in \mathcal{Y}_y} p_z = \sum_{z \in \{y\}} p_z = p_y \qquad (y \in \mathcal{Y}). \tag{27}$$

Define the predicted conditional probability of proceeding to child v_j, conditional on reaching internal node v under the prediction, by

$$q_{v,j}(p) := \frac{\pi_{v_j}(p)}{\pi_v(p)}, \qquad j = 1, \ldots, d_v \tag{28}$$

and write $\boldsymbol{q}_v(p) := (q_{v,1}(p), \ldots, q_{v,d_v}(p))^\top$. By Eq. (21),

$$\sum_{j=1}^{d_v} q_{v,j}(p) = \frac{\sum_{j=1}^{d_v} \pi_{v_j}(p)}{\pi_v(p)} = \frac{\sum_{y \in \mathcal{Y}_v} p_y}{\pi_v(p)} = 1.$$

Therefore $\boldsymbol{q}_v(p) \in \Delta^{d_v - 1}$.

Rearranging Eq. (28) gives, for every leaf y and each $r = 0, \ldots, L_y - 1$,

$$\pi_{w_{r+1}(y)}(p) = \pi_{w_r(y)}(p) q_{w_r(y), j_r(y)}(p) \tag{29}$$

Thus the next node's predicted reach probability equals the current node's predicted reach probability times the selected branch probability. Taking the product in Eq. (28) along the path gives

$$\begin{aligned} \prod_{r=0}^{L_y-1} q_{w_r(y), j_r(y)}(p) &= \prod_{r=0}^{L_y-1} \frac{\pi_{w_{r+1}(y)}(p)}{\pi_{w_r(y)}(p)} \\ &= \frac{\pi_{w_1(y)}(p)}{\pi_{w_0(y)}(p)} \frac{\pi_{w_2(y)}(p)}{\pi_{w_1(y)}(p)} \cdots \frac{\pi_{w_{L_y}(y)}(p)}{\pi_{w_{L_y-1}(y)}(p)} \\ &= \frac{\pi_{w_{L_y}(y)}(p)}{\pi_{w_0(y)}(p)} = \frac{\pi_y(p)}{\pi_\rho(p)} = p_y. \end{aligned} \tag{30}$$

The intermediate reach probabilities $\pi_{w_1(y)}(p), \ldots, \pi_{w_{L_y-1}(y)}(p)$ each appear once in a numerator and once in a denominator and hence cancel. We then use $w_0(y) = \rho$ and $w_{L_y}(y) = y$ from Eq. (23). Finally, substituting $\pi_\rho(p) = 1$ and $\pi_y(p) = p_y$ from Eq. (27) gives the last equality. Thus each predicted leaf probability is the product of the branch probabilities along its path.

Let $u : \Delta^{K-1} \times \mathcal{Y} \to [-1, 1]$ be a utility. For a node a, define the numerator of its subtree mean utility and the subtree mean itself by

$$N_{u,a}(p) := \sum_{y \in \mathcal{Y}_a} p_y u(p, y), \tag{31}$$

$$\mu_{u,a}(p) := \frac{N_{u,a}(p)}{\pi_a(p)} \tag{32}$$

At the root and at a leaf,

$$\mu_{u,\rho}(p) = \sum_{y \in \mathcal{Y}} p_y u(p, y) =: \mu_u(p), \qquad \mu_{u,y}(p) = u(p, y). \tag{33}$$

Define the true conditional reach probability directly from the conditional distribution $\eta(p)$ by

$$\pi^\star_a(p) := \sum_{y \in \mathcal{Y}_a} \eta_y(p) = \mathbb{P}(Y \in \mathcal{Y}_a \mid P = p) \tag{34}$$

This is always defined for every node a. Only when $\pi^\star_v(p) > 0$ do we define the true conditional probability of selecting child v_j, conditional on truly reaching v, by

$$q^\star_{v,j}(p) := \frac{\pi^\star_{v_j}(p)}{\pi^\star_v(p)} \tag{35}$$

If $\pi^\star_v(p) = 0$, the conditioning event $\{Y \in \mathcal{Y}_v\}$ has conditional probability zero, so we do not define $q^\star_{v,j}(p)$. Setting all components to zero would give a vector whose components sum to zero, not a probability vector, so it would also be inappropriate to call that vector the true conditional branch probabilities. Even in this case, $\pi^\star_{v_j}(p) = 0$ for every child v_j. Subsequent proofs use $\pi^\star_{v_j}(p) - \pi^\star_v(p) q_{v,j}(p)$, which remains defined without conditioning on a null event.

With the subtree indicators introduced above, define the local branch residual and local utility contrast componentwise by

$$R_{v,j}(p, y) := J_{v,j}(y) - J_v(y) q_{v,j}(p), \tag{36}$$

$$\Delta_{u,v,j}(p) := \mu_{u,v_j}(p) - \mu_{u,v}(p) \tag{37}$$

and write $\boldsymbol{R}_v(p, y) := (R_{v,1}(p, y), \ldots, R_{v,d_v}(p, y))^\top$ and $\boldsymbol{\Delta}_{u,v}(p) := (\Delta_{u,v,1}(p), \ldots, \Delta_{u,v,d_v}(p))^\top$, respectively.

Let $\mathcal{I} := \{I \subseteq [-1, 1] : I \text{ is an interval}\} \cup \{\varnothing\}$. Let $\mathcal{C}$ be a family of measurable functions $c : \mathcal{X} \to [-1, 1]$, and let $\mathcal{U}$ be a family of measurable utilities $u : \Delta^{K-1} \times \mathcal{Y} \to [-1, 1]$. For a fixed (c, u, I, v), define the node-wise audit moment by

$$\Gamma_{c,u,I,v}(f) := \mathbb{E}[c(X)\mathbf{1}\{\mu_u(P) \in I\} \langle \boldsymbol{\Delta}_{u,v}(P), \boldsymbol{R}_v(P, Y) \rangle] \tag{38}$$

Also set

$$\mathcal{V}_\mathcal{T}(u) := \{v \in V^\circ : \boldsymbol{\Delta}_{u,v} \not\equiv \mathbf{0}\} \tag{39}$$

Here $\boldsymbol{\Delta}_{u,v} \not\equiv \mathbf{0}$ means that, for at least one prediction vector p with positive components, the subtree mean utilities $\mu_{u,v_1}(p), \ldots, \mu_{u,v_{d_v}}(p)$ are not all equal. Define

$$\mathrm{HUC}(f; \mathcal{C}, \mathcal{U}) := \sup_{\substack{c \in \mathcal{C},\, u \in \mathcal{U},\, I \in \mathcal{I} \\ v \in \mathcal{V}_\mathcal{T}(u)}} |\Gamma_{c,u,I,v}(f)| \tag{40}$$

If the candidate set is empty, define the supremum to be zero. For comparison, set

$$\mathrm{UC}(f; \mathcal{C}, \mathcal{U}) := \sup_{c \in \mathcal{C},\, u \in \mathcal{U},\, I \in \mathcal{I}} |\mathbb{E}[c(X)\mathbf{1}\{\mu_u(P) \in I\}\{u(P, Y) - \mu_u(P)\}]| \tag{41}$$

as the corresponding UC error. Whenever $\mathcal{C}$ and $\mathcal{U}$ are finite, write the number of audited triples (c, u, v) as

$$N_{\mathrm{cand}} := |\mathcal{C}| \sum_{u \in \mathcal{U}} |\mathcal{V}_\mathcal{T}(u)|. \tag{42}$$

B Binary Classification and Binary Trees

Before proceeding to general multiclass classification, we first build intuition through a concrete example of binary classification with a binary tree.

B.1 Binary Classification: A Single Branching Node

Let $\mathcal{Y} = \{L, R\}$, let $p_L = 1 - p_R$, and write $q_L(p) := p_L$ and $q_R(p) := p_R$. For utility values $u_L := u(p, L)$ and $u_R := u(p, R)$, the predicted mean utility is

$$\mu_u(p) = q_L(p)u_L + q_R(p)u_R \tag{43}$$

Define the local utility contrast and binary residual by

$$\Delta_u(p) := u_R - u_L, \qquad R(p, y) := \mathbf{1}\{y = R\} - q_R(p) \tag{44}$$

respectively.

Lemma B.1 (Pointwise identity for binary classification)**.** *For every $y \in \{L, R\}$,*

$$u(p, y) - \mu_u(p) = \Delta_u(p)R(p, y). \tag{45}$$

Proof. If $y = R$, then $R(p, R) = 1 - q_R(p) = q_L(p)$. Hence

$$\Delta_u(p)R(p, R) = q_L(p)(u_R - u_L) = u_R - \{q_L(p)u_L + q_R(p)u_R\} = u(p, R) - \mu_u(p).$$

The third equality uses $u_R = \{q_L(p) + q_R(p)\}u_R$.

If $y = L$, then $R(p, L) = -q_R(p)$. Hence

$$\Delta_u(p)R(p, L) = -q_R(p)(u_R - u_L) = u_L - \{q_L(p)u_L + q_R(p)u_R\} = u(p, L) - \mu_u(p).$$

The third equality uses $u_L = \{q_L(p) + q_R(p)\}u_L$. □

Writing the true right-label probability as $q^\star_R(p) := \mathbb{P}(Y = R \mid P = p)$ gives

$$\mathbb{E}[R(p, Y) \mid P = p] = \mathbb{P}(Y = R \mid P = p) - q_R(p) = q^\star_R(p) - q_R(p). \tag{46}$$

Consequently,

$$\mathbb{E}[u(p, Y) - \mu_u(p) \mid P = p] = \{q^\star_R(p) - q_R(p)\}\Delta_u(p). \tag{47}$$

B.2 A Single Internal Node of a Binary Tree

Write the left and right children of an internal node v in a binary tree as v_L, v_R. Expanding the definition of subtree mean utility gives

$$\mu_{u,v}(p) = \frac{\pi_{v_L}(p)\mu_{u,v_L}(p) + \pi_{v_R}(p)\mu_{u,v_R}(p)}{\pi_v(p)} = q_{v,L}(p)\mu_{u,v_L}(p) + q_{v,R}(p)\mu_{u,v_R}(p). \tag{48}$$

Define the local utility contrast and binary branch residual by

$$\Delta_{u,v}(p) := \mu_{u,v_R}(p) - \mu_{u,v_L}(p), \qquad R_v(p, y) := J_{v,R}(y) - q_{v,R}(p)J_v(y) \tag{49}$$

respectively.

Lemma B.2 (Local identity at a binary branching node)**.** *For every leaf y,*

$$\Delta_{u,v}(p)R_v(p, y) = \begin{cases} \mu_{u,v_R}(p) - \mu_{u,v}(p), & y \in \mathcal{Y}_{v_R}, \\ \mu_{u,v_L}(p) - \mu_{u,v}(p), & y \in \mathcal{Y}_{v_L}, \\ 0, & y \notin \mathcal{Y}_v \end{cases}. \tag{50}$$

Proof. If $y \in \mathcal{Y}_{v_R}$, then $J_v(y) = J_{v,R}(y) = 1$, so $R_v(p, y) = 1 - q_{v,R}(p) = q_{v,L}(p)$. Hence

$$\Delta_{u,v}(p)R_v(p, y) = q_{v,L}(p)\{\mu_{u,v_R}(p) - \mu_{u,v_L}(p)\} = \mu_{u,v_R}(p) - \mu_{u,v}(p).$$

The last equality uses Eq. (48).

If $y \in \mathcal{Y}_{v_L}$, then $J_v(y) = 1$ and $J_{v,R}(y) = 0$, so $R_v(p, y) = -q_{v,R}(p)$. Hence

$$\Delta_{u,v}(p)R_v(p, y) = -q_{v,R}(p)\{\mu_{u,v_R}(p) - \mu_{u,v_L}(p)\} = \mu_{u,v_L}(p) - \mu_{u,v}(p).$$

If $y \notin \mathcal{Y}_v$, then $J_v(y) = J_{v,R}(y) = 0$ and therefore $R_v(p, y) = 0$. □

B.3 Path Decomposition and Conditional Means on a Binary Tree

Proposition B.3 (Utility-error decomposition on a binary tree)**.** *Suppose every internal node has two children. For every p and leaf y,*

$$u(p,y) - \mu_u(p) = \sum_{v \in V^\circ} \Delta_{u,v}(p) R_v(p,y). \tag{51}$$

Proof. Write the path to leaf y as $\rho = w_0, \dots, w_{L_y} = y$. For every r, w_{r+1} is either the left or the right child of w_r. By Lemma B.2, $\Delta_{u,w_r}(p) R_{w_r}(p,y) = \mu_{u,w_{r+1}}(p) - \mu_{u,w_r}(p)$. Summing along the path gives

$$\begin{aligned}\sum_{r=0}^{L_y-1} \Delta_{u,w_r}(p) R_{w_r}(p,y) &= \sum_{r=0}^{L_y-1} \{\mu_{u,w_{r+1}}(p) - \mu_{u,w_r}(p)\} \\ &= \mu_{u,y}(p) - \mu_{u,\rho}(p) = u(p,y) - \mu_u(p).\end{aligned}$$

Each intermediate subtree mean utility occurs once with a positive sign and once with a negative sign and hence cancels. For a node v outside the path, $y \notin \mathcal{Y}_v$, so $R_v(p,y) = 0$. The sum along the path can therefore be extended to all internal nodes. □

For an internal node v of a binary tree, the true conditional reach probability of its right subtree is $\pi^\star_{v_R}(p) = \mathbb{P}(Y \in \mathcal{Y}_{v_R} \mid P = p)$. Thus

$$\begin{aligned}\mathbb{E}[R_v(p,Y) \mid P = p] &= \mathbb{P}(Y \in \mathcal{Y}_{v_R} \mid P = p) - q_{v,R}(p)\mathbb{P}(Y \in \mathcal{Y}_v \mid P = p) \\ &= \pi^\star_{v_R}(p) - q_{v,R}(p)\pi^\star_v(p).\end{aligned} \tag{52}$$

Since $\Delta_{u,v}(p)$ is constant conditional on $P = p$,

$$\mathbb{E}[\Delta_{u,v}(p) R_v(p,Y) \mid P = p] = \{\pi^\star_{v_R}(p) - q_{v,R}(p)\pi^\star_v(p)\}\Delta_{u,v}(p). \tag{53}$$

When $\pi^\star_v(p) > 0$, we can define $q^\star_v(p) := \pi^\star_{v_R}(p)/\pi^\star_v(p)$ and rewrite the right-hand side as

$$\pi^\star_v(p)\{q^\star_v(p) - q_{v,R}(p)\}\Delta_{u,v}(p).$$

When $\pi^\star_v(p) = 0$, we also have $\pi^\star_{v_R}(p) = 0$. The right-hand side of Eq. (53) is then already zero, without defining $q^\star_v(p)$.

B.4 Four-Leaf Mean-Utility Decomposition in Section 2.2

This subsection verifies the child-utility notation and proves Eq. (5) directly, without using the general-tree decomposition. Fix the four-leaf tree, a positive prediction p, and the target utility u of Section 2.2. Use the conditional distribution $\eta_y(p)$ from Appendix A.2; conditional identities hold for almost every p under the law of P. Assume $0 < q^\star_{0,R} < 1$ as in that section.

Child means and their predicted averages. For $t \in \{0, L, R\}$ and $j \in \{L, R\}$, the child mean is

$$\bar{u}_{t,j} := \sum_{y \in \mathcal{Y}_{v_{t,j}}} p_y u(p,y) \,/ \sum_{y \in \mathcal{Y}_{v_{t,j}}} p_y = \mu_{u,v_{t,j}}(p). \tag{54}$$

The denominator is positive, and the normalized weights are nonnegative and sum to one. Thus $\bar{u}_{t,j} \in [-1,1]$ and, for $\widetilde{Y} \sim p$, $\bar{u}_{t,j} = \mathbb{E}_{\widetilde{Y} \sim p}[u(p,\widetilde{Y}) \mid \widetilde{Y} \in \mathcal{Y}_{v_{t,j}}]$. These means are measurable functions of p, since they are ratios of finite sums of measurable functions with positive denominators. As in Section 2.2, dependence on the fixed u, p is suppressed. The lower children are leaves: $(v_{L,L}, v_{L,R}, v_{R,L}, v_{R,R}) = (y_1, y_2, y_3, y_4)$. Hence $\bar{u}_{t,j} = u(p, v_{t,j})$ for $t \in \{L, R\}$.

For $t, j \in \{L, R\}$, at the root $q_{0,t} = \sum_{y \in \mathcal{Y}_{v_t}} p_y$, and at a lower node $q_{t,j} = p_{v_{t,j}}/q_{0,t}$. Therefore

$$\begin{aligned}\sum_{j=L,R} q_{t,j}\bar{u}_{t,j} &= \sum_{j=L,R} (p_{v_{t,j}}/q_{0,t}) u(p, v_{t,j}) = \bar{u}_{0,t}, \\ \sum_{t=L,R} q_{0,t}\bar{u}_{0,t} &= \sum_{t=L,R} \sum_{y \in \mathcal{Y}_{v_t}} p_y u(p,y) = \mu_u(p).\end{aligned} \tag{55}$$

The first identity holds for $t \in \{L, R\}$. It expresses the predicted mean at a parent as the predicted-branch average of its child means.

Proposition B.4 (Four-leaf mean-utility decomposition)**.** *With the notation of Section 2.2,* $D^{\mathrm{full}}(p) = D_{\mathrm{top}}(p) + D_L(p) + D_R(p)$ *as in Eq.* (5)*. Each term is the conditional mean of the utility increment at its corresponding node.*

Proof. For $t, j \in \{L, R\}$, the multiplication rule for conditional probabilities gives $\mathbb{P}(Y = v_{t,j} \mid P = p) = q^\star_{0,t} q^\star_{t,j}$. Because these children are leaves, the true conditional mean is

$$\mathbb{E}[u(p, Y) \mid P = p] = \sum_{t=L,R} q^\star_{0,t} \sum_{j=L,R} q^\star_{t,j} \bar{u}_{t,j}. \tag{56}$$

Subtract the predicted root mean in Eq. (55) and add and subtract $\sum_{t=L,R} q^\star_{0,t} \bar{u}_{0,t}$:

$$\begin{aligned} D^{\mathrm{full}}(p) &= \sum_{t=L,R} q^\star_{0,t} \sum_{j=L,R} q^\star_{t,j} \bar{u}_{t,j} - \sum_{t=L,R} q_{0,t} \bar{u}_{0,t} \\ &= [\sum_{t=L,R} q^\star_{0,t} \bar{u}_{0,t} - \sum_{t=L,R} q_{0,t} \bar{u}_{0,t}] + \sum_{t=L,R} q^\star_{0,t} [\sum_{j=L,R} q^\star_{t,j} \bar{u}_{t,j} - \bar{u}_{0,t}] \\ &= D_{\mathrm{top}}(p) + \sum_{t=L,R} q^\star_{0,t} [\sum_{j=L,R} q^\star_{t,j} \bar{u}_{t,j} - \sum_{j=L,R} q_{t,j} \bar{u}_{t,j}] \\ &= D_{\mathrm{top}}(p) + D_L(p) + D_R(p). \end{aligned}$$

The third equality uses the first identity in Eq. (55).

For the increment interpretation, consider an observed leaf $y = v_{t,j}$. Along its two-step path,

$$u(p, y) - \mu_u(p) = \{\bar{u}_{0,t} - \mu_u(p)\} + \{\bar{u}_{t,j} - \bar{u}_{0,t}\}.$$

Equivalently, using the subtree indicators of Appendix B,

$$u(p, y) - \mu_u(p) = \sum_{t=L,R} J_{v_t}(y) \{\bar{u}_{0,t} - \mu_u(p)\} + \sum_{t=L,R} \sum_{j=L,R} J_{v_{t,j}}(y) \{\bar{u}_{t,j} - \bar{u}_{0,t}\}. \tag{57}$$

The conditional mean of the root term is

$$\sum_{t=L,R} q^\star_{0,t} \{\bar{u}_{0,t} - \mu_u(p)\} = \sum_{t=L,R} q^\star_{0,t} \bar{u}_{0,t} - \sum_{t=L,R} q_{0,t} \bar{u}_{0,t} = D_{\mathrm{top}}(p).$$

For a lower node v_t, its conditional mean is

$$\sum_{j=L,R} q^\star_{0,t} q^\star_{t,j} \{\bar{u}_{t,j} - \bar{u}_{0,t}\} = q^\star_{0,t} [\sum_{j=L,R} q^\star_{t,j} \bar{u}_{t,j} - \sum_{j=L,R} q_{t,j} \bar{u}_{t,j}] = D_t(p).$$

Here $\sum_j q^\star_{t,j} = 1$ and Eq. (55) give the equality. Thus the lower-node contributions retain their true reach probabilities. □

Correspondence with the binary-node calculation. Since $q^\star_{t,L} - q_{t,L} = -(q^\star_{t,R} - q_{t,R})$, the same terms can also be written as

$$\begin{aligned} D_{\mathrm{top}}(p) &= (q^\star_{0,R} - q_{0,R})(\bar{u}_{0,R} - \bar{u}_{0,L}), \\ D_t(p) &= q^\star_{0,t} (q^\star_{t,R} - q_{t,R})(\bar{u}_{t,R} - \bar{u}_{t,L}), \qquad t \in \{L, R\}. \end{aligned} \tag{58}$$

These are precisely the conditional means in Appendix B.3: the binary contrast is the right-child mean minus the left-child mean. By Lemma C.2, the corresponding summands in Eq. (57) also equal the vector inner products used in Section 3.1. This verifies the correspondence between the three quantities in Eq. (5) and the general-node contributions.

The UC identity in Section 2.2. For a predictor satisfying the four-leaf assumptions almost surely, write $D^{\mathrm{full}}(P) = \mathbb{E}[u(P, Y) - \mu_u(P) \mid P]$ and restore the dependence of the D_t on P. For any $I \in \mathcal{I}$, the interval indicator is measurable with respect to P, so

$$\begin{aligned} &\mathbb{E}[\mathbf{1}\{\mu_u(P) \in I\} \{u(P, Y) - \mu_u(P)\}] = \mathbb{E}[\mathbf{1}\{\mu_u(P) \in I\} D^{\mathrm{full}}(P)] \\ &= \sum_{t \in \{\mathrm{top}, L, R\}} \mathbb{E}[\mathbf{1}\{\mu_u(P) \in I\} D_t(P)]. \end{aligned} \tag{59}$$

Taking the absolute value and then the supremum over I gives the displayed UC identity in Section 2.2. Thus the sum is taken before the absolute value, allowing opposite node contributions to cancel within the same predicted-utility interval.

C Extension to General Trees

For binary trees, each internal node was associated with one scalar residual and one scalar utility contrast. At a general internal node v with d_v children, both the local branch residual and the local utility contrast are d_v-dimensional vectors. We use the predicted reach probability $\pi_a(p)$, predicted conditional branch probability $q_{v,j}(p)$, and path-product representation already defined and derived in Eq. (26), Eq. (28), and Eq. (30). No additional auxiliary stochastic process is introduced; the argument uses only these quantities determined by the leaf distribution p.

C.1 Local Algebra on a General Tree

Lemma C.1 (Recursions for subtree probabilities and subtree mean utilities)**.** *For every internal node v,*

$$\pi_v(p) = \sum_{j=1}^{d_v} \pi_{v_j}(p), \tag{60}$$

$$N_{u,v}(p) = \sum_{j=1}^{d_v} N_{u,v_j}(p), \tag{61}$$

$$\mu_{u,v}(p) = \sum_{j=1}^{d_v} q_{v,j}(p)\mu_{u,v_j}(p). \tag{62}$$

Moreover,

$$\langle \boldsymbol{q}_v(p), \boldsymbol{\Delta}_{u,v}(p) \rangle = 0. \tag{63}$$

Proof. By Eq. (21),

$$\pi_v(p) = \sum_{y \in \mathcal{Y}_v} p_y = \sum_{j=1}^{d_v} \sum_{y \in \mathcal{Y}_{v_j}} p_y = \sum_{j=1}^{d_v} \pi_{v_j}(p),$$

$$N_{u,v}(p) = \sum_{y \in \mathcal{Y}_v} p_y u(p,y) = \sum_{j=1}^{d_v} \sum_{y \in \mathcal{Y}_{v_j}} p_y u(p,y) = \sum_{j=1}^{d_v} N_{u,v_j}(p).$$

Since $N_{u,v_j}(p) = \pi_{v_j}(p)\mu_{u,v_j}(p)$,

$$\mu_{u,v}(p) = \frac{\sum_{j=1}^{d_v} \pi_{v_j}(p)\mu_{u,v_j}(p)}{\pi_v(p)} = \sum_{j=1}^{d_v} q_{v,j}(p)\mu_{u,v_j}(p).$$

Finally,

$$\langle \boldsymbol{q}_v(p), \boldsymbol{\Delta}_{u,v}(p) \rangle = \sum_{j=1}^{d_v} q_{v,j}(p)\mu_{u,v_j}(p) - \mu_{u,v}(p) \sum_{j=1}^{d_v} q_{v,j}(p) = 0.$$

□

Lemma C.2 (Local contribution of one node)**.** *If $y \in \mathcal{Y}_{v_k}$, then*

$$\langle \boldsymbol{\Delta}_{u,v}(p), \boldsymbol{R}_v(p,y) \rangle = \mu_{u,v_k}(p) - \mu_{u,v}(p). \tag{64}$$

If $y \notin \mathcal{Y}_v$, the left-hand side is zero. Moreover, for every p, y,

$$\sum_{j=1}^{d_v} R_{v,j}(p,y) = 0. \tag{65}$$

Proof. Suppose $y \in \mathcal{Y}_{v_k}$. Then $J_v(y) = 1$, $J_{v,k}(y) = 1$, and $J_{v,j}(y) = 0$ for $j \neq k$. Thus

$$R_{v,j}(p,y) = \mathbf{1}\{j = k\} - q_{v,j}(p). \tag{66}$$

Consequently,

$$\langle \boldsymbol{\Delta}_{u,v}(p), \boldsymbol{R}_v(p,y) \rangle = \Delta_{u,v,k}(p) - \sum_{j=1}^{d_v} q_{v,j}(p) \Delta_{u,v,j}(p) = \mu_{u,v_k}(p) - \mu_{u,v}(p).$$

The last equality uses Eq. (63).

If $y \notin \mathcal{Y}_v$, then $J_v(y) = J_{v,j}(y) = 0$ for every j, so $\boldsymbol{R}_v(p,y) = \mathbf{0}$. Furthermore,

$$\sum_{j=1}^{d_v} R_{v,j}(p,y) = \sum_{j=1}^{d_v} J_{v,j}(y) - J_v(y) \sum_{j=1}^{d_v} q_{v,j}(p) = 0.$$

□

C.2 Utility-Error Decomposition and Conditional Means on a General Tree

Theorem C.3 (Utility-error decomposition on a general tree)**.** *For every positive prediction vector p and every leaf y,*

$$u(p,y) - \mu_u(p) = \sum_{v \in V^\circ} \langle \boldsymbol{\Delta}_{u,v}(p), \boldsymbol{R}_v(p,y) \rangle . \tag{67}$$

Proof. For a leaf y, use the path sequence defined in Eq. (22), abbreviating $w_r := w_r(y)$ in this proof. By Lemma C.2, each internal node w_r along the path satisfies $\langle \boldsymbol{\Delta}_{u,w_r}(p), \boldsymbol{R}_{w_r}(p,y) \rangle = \mu_{u,w_{r+1}}(p) - \mu_{u,w_r}(p)$. Therefore,

$$\begin{aligned} \sum_{r=0}^{L_y-1} \langle \boldsymbol{\Delta}_{u,w_r}(p), \boldsymbol{R}_{w_r}(p,y) \rangle &= \sum_{r=0}^{L_y-1} \{\mu_{u,w_{r+1}}(p) - \mu_{u,w_r}(p)\} \\ &= \mu_{u,y}(p) - \mu_{u,\rho}(p) = u(p,y) - \mu_u(p). \end{aligned}$$

At an internal node v outside the path, $y \notin \mathcal{Y}_v$, so its local contribution is zero by Lemma C.2. Hence the sum along the path can be extended to all of V°. □

Lemma C.4 (Conditional mean of a branch residual and mean-utility matching)**.** *For every internal node v and child v_j,*

$$\mathbb{E}[R_{v,j}(p,Y) \mid P = p] = \pi^\star_{v_j}(p) - \pi^\star_v(p) q_{v,j}(p). \tag{68}$$

Consequently, the conditional mean of the local contribution of v is

$$\begin{aligned} \mathbb{E}[\langle \boldsymbol{\Delta}_{u,v}(p), \boldsymbol{R}_v(p,Y) \rangle \mid P = p] &= \sum_{j=1}^{d_v} \Delta_{u,v,j}(p) \{\pi^\star_{v_j}(p) - \pi^\star_v(p) q_{v,j}(p)\} \\ = \sum_{j=1}^{d_v} \pi^\star_{v_j}(p) \mu_{u,v_j}(p) - \pi^\star_v(p) \mu_{u,v}(p). \end{aligned} \tag{69}$$

If $\pi^\star_v(p) > 0$, this is equivalently

$$\mathbb{E}[\langle \boldsymbol{\Delta}_{u,v}(p), \boldsymbol{R}_v(p,Y) \rangle \mid P = p] = \pi^\star_v(p) [\sum_{j=1}^{d_v} q^\star_{v,j}(p) \mu_{u,v_j}(p) - \sum_{j=1}^{d_v} q_{v,j}(p) \mu_{u,v_j}(p)]. \tag{70}$$

If $\pi^\star_v(p) = 0$, the quantities in Eq. (69) *are zero without defining $q^\star_{v,j}(p)$.*

Proof. Taking conditional expectations in Eq. (36) gives

$$\begin{aligned} \mathbb{E}[R_{v,j}(p,Y) \mid P = p] &= \mathbb{E}[J_{v,j}(Y) \mid P = p] - q_{v,j}(p) \mathbb{E}[J_v(Y) \mid P = p] \\ &= \pi^\star_{v_j}(p) - \pi^\star_v(p) q_{v,j}(p). \end{aligned}$$

This proves Eq. (68). The utility contrast is fixed conditional on $P = p$; expanding the inner product and interchanging the finite sum with conditional expectation gives the first equality in Eq. (69).

The true child subtrees partition the parent subtree, so $\sum_j \pi^\star_{v_j}(p) = \pi^\star_v(p)$, and $\sum_j q_{v,j}(p) = 1$. Substituting $\Delta_{u,v,j}(p) = \mu_{u,v_j}(p) - \mu_{u,v}(p)$ and using these identities, the parent-mean terms cancel:

$$\begin{aligned}\sum_{j=1}^{d_v} \Delta_{u,v,j}(p)\{\pi^\star_{v_j}(p) - \pi^\star_v(p)q_{v,j}(p)\} &= \sum_{j=1}^{d_v} \mu_{u,v_j}(p)\{\pi^\star_{v_j}(p) - \pi^\star_v(p)q_{v,j}(p)\} \\ &= \sum_{j=1}^{d_v} \pi^\star_{v_j}(p)\mu_{u,v_j}(p) - \pi^\star_v(p)\sum_{j=1}^{d_v} q_{v,j}(p)\mu_{u,v_j}(p) \\ &= \sum_{j=1}^{d_v} \pi^\star_{v_j}(p)\mu_{u,v_j}(p) - \pi^\star_v(p)\mu_{u,v}(p).\end{aligned}$$

The last line uses Eq. (62). This proves the second equality in Eq. (69) for every true reach probability.

If $\pi^\star_v(p) > 0$, substitute $\pi^\star_{v_j}(p) = \pi^\star_v(p)q^\star_{v,j}(p)$ in the penultimate line and factor out $\pi^\star_v(p)$. This gives Eq. (70) and Eq. (8). If $\pi^\star_v(p) = 0$, then $0 \le \pi^\star_{v_j}(p) \le \pi^\star_v(p) = 0$ for every child, so all expressions in Eq. (69) vanish without conditioning on a null event. □

Meaning of the two averages. For $\widetilde{Y} \sim p$, each value $\mu_{u,v_j}(p)$ is the conditional average of $u(p, \widetilde{Y})$ given $\widetilde{Y} \in \mathcal{Y}_{v_j}$. Thus the finer outcomes within a child are marginalized under the predictive distribution. When $\pi^\star_v(p) > 0$, the actual child index $j_v(Y)$, conditional on $P = p$ and $Y \in \mathcal{Y}_v$, has probabilities $q^\star_{v,j}(p)$. Consequently,

$$\mathbb{E}[\mu_{u,v_{j_v(Y)}}(p) - \mu_{u,v}(p) \mid P = p,\ Y \in \mathcal{Y}_v] = \sum_{j=1}^{d_v} q^\star_{v,j}(p)\mu_{u,v_j}(p) - \sum_{j=1}^{d_v} q_{v,j}(p)\mu_{u,v_j}(p). \quad (71)$$

Only the outer branch probabilities differ between the two sums; both use the same predictively marginalized child values. Equation (70) multiplies this difference by the actual probability of reaching the parent. Under the predictive distribution instead, the child average equals $\mu_{u,v}(p)$ by Eq. (62).

Corollary C.5 (Node decomposition of the cell-wise UC residual)**.** *For every c, u, I,*

$$\mathbb{E}[c(X)\mathbf{1}\{\mu_u(P) \in I\}\{u(P,Y) - \mu_u(P)\}] = \sum_{v \in V^\circ} \Gamma_{c,u,I,v}(f). \quad (72)$$

Proof. Substitute $p = P$ and $y = Y$ into Theorem C.3, multiply both sides by $c(X)\mathbf{1}\{\mu_u(P) \in I\}$, and take expectations. Since the tree is finite, the expectation and the finite sum can be interchanged:

$$\begin{aligned}&\mathbb{E}[c(X)\mathbf{1}\{\mu_u(P) \in I\}\{u(P,Y) - \mu_u(P)\}] \\ &= \sum_{v \in V^\circ} \mathbb{E}[c(X)\mathbf{1}\{\mu_u(P) \in I\}\,\langle \boldsymbol{\Delta}_{u,v}(P), \boldsymbol{R}_v(P,Y)\rangle] \\ &= \sum_{v \in V^\circ} \Gamma_{c,u,I,v}(f).\end{aligned}$$ □

Mean consistency under common audit conditions. Lemma C.2 and $\sum_j J_{v,j}(y) = J_v(y)$ give the pointwise identity

$$\langle \boldsymbol{\Delta}_{u,v}(p), \boldsymbol{R}_v(p,y)\rangle = \sum_{j=1}^{d_v} J_{v,j}(y)\mu_{u,v_j}(p) - J_v(y)\mu_{u,v}(p).$$

Multiplying by the common audit weight and taking expectation under the actual joint law yields

$$\Gamma_{c,u,I,v}(f) = \mathbb{E}\left[c(X)\mathbf{1}\{\mu_u(P) \in I\}\{\sum_{j=1}^{d_v} J_{v,j}(Y)\mu_{u,v_j}(P) - J_v(Y)\mu_{u,v}(P)\}\right]. \quad (73)$$

The expression in braces is the child's utility assessment minus the parent's on reaching v, and zero otherwise. Hence $|\Gamma_{c,u,I,v}(f)| \le \mathrm{HUC}(f;\mathcal{C},\mathcal{U})$ for every $c \in \mathcal{C}$, $u \in \mathcal{U}$, $I \in \mathcal{I}$, and relevant v; irrelevant nodes have zero contribution. This is the reach-weighted, audit-moment sense in which HUC evaluates mean consistency under hierarchical refinement uniformly across nodes. The identity retains $c(X)$ inside the expectation, so it applies to the full subgroup class in the definition of HUC.

C.3 Orthogonality and Uniqueness of the Node Decomposition

The telescoping argument proves that the node contributions sum to the utility residual. We now show that, for a fixed tree and predictive distribution, the same decomposition is an orthogonal expansion and is therefore unique and nonredundant.

Fix a prediction vector p with strictly positive coordinates. For functions $h, g : \mathcal{Y} \to \mathbb{R}$, define

$$\langle h, g\rangle_p := \sum_{y\in\mathcal{Y}} p_y h(y) g(y), \qquad \|h\|_p^2 := \langle h, h\rangle_p. \tag{74}$$

For each internal node v, choose vectors $a_{v,1}, \dots, a_{v,d_v-1} \in \mathbb{R}^{d_v}$ satisfying

$$\sum_{j=1}^{d_v} q_{v,j}(p) a_{v,\ell,j} = 0, \qquad \sum_{j=1}^{d_v} q_{v,j}(p) a_{v,\ell,j} a_{v,m,j} = \mathbf{1}\{\ell = m\}. \tag{75}$$

Thus these vectors form an orthonormal basis, under the $\boldsymbol{q}_v(p)$-weighted inner product, for the $(d_v - 1)$-dimensional centered child space at v. Lift each vector to the leaves by

$$\phi_{v,\ell,p}(y) := \begin{cases} \frac{a_{v,\ell,j}}{\sqrt{\pi_v(p)}}, & y \in \mathcal{Y}_{v_j}, \\ 0, & y \notin \mathcal{Y}_v, \end{cases} \qquad \ell = 1, \dots, d_v - 1. \tag{76}$$

Proposition C.6 (Probability-weighted orthogonal representation)**.** *The collection $\{\phi_{v,\ell,p} : v \in V^\circ,\ \ell = 1, \dots, d_v - 1\}$ is an orthonormal basis, under the inner product in Eq.* (74)*, for the leaf functions having zero p-mean. Its size is*

$$\sum_{v\in V^\circ} (d_v - 1) = K - 1. \tag{77}$$

For a target utility u, define

$$\theta_{u,v,\ell}(p) := \sqrt{\pi_v(p)} \sum_{j=1}^{d_v} q_{v,j}(p) \Delta_{u,v,j}(p) a_{v,\ell,j}. \tag{78}$$

Then the contribution of node v and the complete centered utility satisfy

$$\langle \boldsymbol{\Delta}_{u,v}(p), \boldsymbol{R}_v(p,y)\rangle = \sum_{\ell=1}^{d_v-1} \theta_{u,v,\ell}(p) \phi_{v,\ell,p}(y), \tag{79}$$

$$u(p,y) - \mu_u(p) = \sum_{v\in V^\circ} \sum_{\ell=1}^{d_v-1} \theta_{u,v,\ell}(p) \phi_{v,\ell,p}(y). \tag{80}$$

This expansion is unique, and the contribution assigned to each node is the unique least-squares component that is zero outside $\mathcal{Y}_v$, constant on each child subtree, and centered under the predictive distribution within $\mathcal{Y}_v$. In particular,

$$\mathrm{Var}_{\widetilde{Y}\sim p}(u(p,\widetilde{Y})) = \sum_{v\in V^\circ} \sum_{\ell=1}^{d_v-1} \theta_{u,v,\ell}(p)^2 = \sum_{v\in V^\circ} \pi_v(p) \sum_{j=1}^{d_v} q_{v,j}(p) \{\mu_{u,v_j}(p) - \mu_{u,v}(p)\}^2. \tag{81}$$

Proof. Eq. (75) and the definition of $\phi_{v,\ell,p}$ imply that every $\phi_{v,\ell,p}$ has zero p-mean and unit norm, and that functions attached to the same node are mutually orthogonal. For distinct nodes, their descendant-leaf sets are either disjoint or one is contained in a child subtree of the other. In the first

case their supports are disjoint. In the second case the function attached to the ancestor is constant on the descendant subtree, whereas the function attached to the descendant has zero p-weighted sum there. Hence the functions are again orthogonal.

Their number is given by Eq. (77), because $\sum_{v\in V^\circ} d_v = |V|-1 = |V^\circ|+K-1$. The zero-$p$-mean functions on K leaves form a $(K-1)$-dimensional space, so this orthonormal collection is a basis.

By Eq. (63), the child-value vector $\boldsymbol{\Delta}_{u,v}(p)$ has zero $\boldsymbol{q}_v(p)$-weighted mean. Expansion in the basis in Eq. (75) and lifting via Eq. (76) give Eq. (79). Summing and using Theorem C.3 give Eq. (80).

Orthogonality makes this representation unique. More explicitly, let $g_u^p(y) := u(p,y) - \mu_u(p)$ and $r_{u,v}^p(y) := \langle \boldsymbol{\Delta}_{u,v}(p), \boldsymbol{R}_v(p,y)\rangle$. For any other branch-local centered function h associated with node v, all orthogonal components of $g_u^p - r_{u,v}^p$ belong to other nodes and are orthogonal to $h - r_{u,v}^p$. Therefore

$$\|g_u^p - h\|_p^2 = \|g_u^p - r_{u,v}^p\|_p^2 + \|r_{u,v}^p - h\|_p^2,$$

which proves the stated least-squares property and its uniqueness. Finally, Parseval's identity gives the first equality in Eq. (81); the second follows from

$$\|r_{u,v}^p\|_p^2 = \pi_v(p)\sum_{j=1}^{d_v} q_{v,j}(p)\{\mu_{u,v_j}(p) - \mu_{u,v}(p)\}^2.$$

□

For a binary tree, $d_v - 1 = 1$, so each internal node contributes one detail coordinate. With the scalar binary residual R_v of Appendix B.2, choosing the standard normalization gives

$$\phi_{v,p}(y) = \frac{R_v(p,y)}{\sqrt{\pi_v(p)q_{v,L}(p)q_{v,R}(p)}}. \tag{82}$$

Thus a binary tree has one probability-weighted orthogonal coordinate per internal node, and Eq. (80) is the corresponding expansion of the centered utility.

Remark C.7 (Reference measure and scope of the orthogonality). The inner product in Eq. (74) uses the model distribution $\widetilde{Y} \sim p$. Orthogonality, uniqueness, and the Parseval identity are therefore statements about how the model decomposes a centered leaf utility. HUC moments, in contrast, average the same node coordinates under the actual joint law of (X, Y). Consequently, this orthogonality does not imply that population HUC moments vanish or cannot cancel after expectations are taken under the true distribution. It gives a nonredundant node decomposition of this pointwise residual.

This characterization is conditional on the chosen label tree and predictive reference distribution p. It establishes uniqueness and least-squares optimality of the node contributions for that tree and p; it does not assert that the tree itself, or the outer ℓ_∞ aggregation used by HUC, is optimal.

D Relationships among UC, HUC, and Canonical Calibration

D.1 UC–HUC Comparison and the Implication of Canonical Calibration

Proposition D.1 (UC–HUC comparison)**.** *For every subgroup class $\mathcal{C}$ and utility u,*

$$\mathrm{UC}(f;\mathcal{C},\{u\}) \le |\mathcal{V}_{\mathcal{T}}(u)|\,\mathrm{HUC}(f;\mathcal{C},\{u\}). \tag{83}$$

Moreover, if f is canonically calibrated, then

$$\mathrm{HUC}(f;\{1\},\mathcal{U}) = 0. \tag{84}$$

Proof. Fix $c \in \mathcal{C}$ and $I \in \mathcal{I}$. By Eq. (72) and the triangle inequality,

$$\begin{aligned}
&|\mathbb{E}[c(X)\mathbf{1}\{\mu_u(P)\in I\}\{u(P,Y)-\mu_u(P)\}]| \\
&= \left|\sum_{v\in V^\circ}\Gamma_{c,u,I,v}(f)\right| = \left|\sum_{v\in\mathcal{V}_{\mathcal{T}}(u)}\Gamma_{c,u,I,v}(f)\right| \le \sum_{v\in\mathcal{V}_{\mathcal{T}}(u)}|\Gamma_{c,u,I,v}(f)| \le |\mathcal{V}_{\mathcal{T}}(u)|\,\mathrm{HUC}(f;\mathcal{C},\{u\}).
\end{aligned}$$

The restriction uses $\mathbf{\Delta}_{u,v} \equiv \mathbf{0}$ for $v \notin \mathcal{V}_{\mathcal{T}}(u)$; taking the supremum over c, I proves Eq. (83).

Recall that $J_a(y) = \mathbf{1}\{y \in \mathcal{Y}_a\}$ indicates membership in the descendant-leaf set of node a, and $J_{v,j}(y) = \mathbf{1}\{y \in \mathcal{Y}_{v_j}\}$ for a child v_j. Next assume canonical calibration: $\mathbb{P}(Y = y \mid P) = P_y$ for every y. At each node a,

$$\mathbb{E}[J_a(Y) \mid P] = \mathbb{P}(Y \in \mathcal{Y}_a \mid P) = \sum_{y \in \mathcal{Y}_a} \mathbb{P}(Y = y \mid P) = \sum_{y \in \mathcal{Y}_a} P_y = \pi_a(P).$$

Thus, for every internal node v and child v_j,

$$\begin{aligned}\mathbb{E}[R_{v,j}(P, Y) \mid P] &= \mathbb{E}[J_{v,j}(Y) \mid P] - q_{v,j}(P)\mathbb{E}[J_v(Y) \mid P] \\ &= \pi_{v_j}(P) - q_{v,j}(P)\pi_v(P) = \pi_{v_j}(P) - \frac{\pi_{v_j}(P)}{\pi_v(P)}\pi_v(P) = 0.\end{aligned}$$

Hence $\mathbb{E}[\boldsymbol{R}_v(P, Y) \mid P] = \mathbf{0}$. The tower property gives

$$\begin{aligned}\Gamma_{1,u,I,v}(f) &= \mathbb{E}[\mathbf{1}\{\mu_u(P) \in I\}\, \langle \mathbf{\Delta}_{u,v}(P), \boldsymbol{R}_v(P, Y)\rangle] \\ &= \mathbb{E}[\mathbf{1}\{\mu_u(P) \in I\}\, \langle \mathbf{\Delta}_{u,v}(P), \mathbb{E}[\boldsymbol{R}_v(P, Y) \mid P]\rangle] = 0.\end{aligned}$$

Since this holds for every u, I, v, it proves Eq. (84). □

Example D.2 (Exact cancellation with zero UC and nonzero HUC). On the four-leaf tree of Section 2.2, consider the constant predictor $P \equiv p = (1/4, 1/4, 1/4, 1/4)$. Let $(u(p, y_1), u(p, y_2), u(p, y_3), u(p, y_4)) = (-1, 1, 1, 1)$ and let the true conditional leaf distribution be $\eta(p) = (1/4, 0, 3/8, 3/8)$. Extend the utility away from this prediction vector to any bounded measurable function. The predicted branch probabilities are $q_{0,R} = q_{L,R} = q_{R,R} = 1/2$, whereas the true branch probabilities are $q^\star_{0,R} = 3/4, q^\star_{L,R} = 0, q^\star_{R,R} = 1/2$, and $\bar{u}_{0,L} = 0, \bar{u}_{0,R} = 1$. Consequently, $D_{\text{top}}(p) = 1/4$, $D_L(p) = -1/4$, $D_R(p) = 0$. The predicted and actual mean utilities are both $1/2$, so the total residual vanishes. An interval either contains the sole score $1/2$ or does not. Thus $\mathrm{UC}(f; \{1\}, \{u\}) = 0$ and $\mathrm{HUC}(f; \{1\}, \{u\}) = 1/4$. This realizes exact cancellation under a valid joint distribution.

Lemma D.3 (Characterization of canonical calibration by branch residuals)**.** *Assume that every component of the prediction vector is positive. The following are equivalent.*

(i) f is canonically calibrated.

(ii) $\mathbb{E}[\boldsymbol{R}_v(P, Y) \mid P] = \mathbf{0}$ for every internal node v.

Proof. The implication from (i) to (ii) was shown in the preceding proof. We now prove the converse. For every node a, write $\zeta_a(P) := \mathbb{E}[J_a(Y) \mid P]$. At the root, $J_\rho(Y) = 1$, so $\zeta_\rho(P) = 1 = \pi_\rho(P)$.

Suppose $\zeta_v(P) = \pi_v(P)$ holds at an internal node v. The jth component of (ii) gives

$$0 = \mathbb{E}[R_{v,j}(P, Y) \mid P] = \zeta_{v_j}(P) - q_{v,j}(P)\zeta_v(P),$$

$$\zeta_{v_j}(P) = q_{v,j}(P)\zeta_v(P) = q_{v,j}(P)\pi_v(P) = \frac{\pi_{v_j}(P)}{\pi_v(P)}\pi_v(P) = \pi_{v_j}(P).$$

Induction from the root to the leaves gives $\zeta_y(P) = \pi_y(P)$ for every leaf y. Since $\zeta_y(P) = \mathbb{P}(Y = y \mid P)$ and $\pi_y(P) = P_y$ at a leaf, canonical calibration follows. □

D.2 Marginal Calibration and Conditional Branches

This example supplements the distinction in Section 2.2 between calibration of individual leaf or subtree probabilities and agreement of conditional branch probabilities. We first verify that the former calibration errors are zero, and then calculate the conditional branch-probability errors that remain.

Consider a tree with leaves $\{e, f, o_1, o_2\}$, in which $D := \{e, f\}$ branches into $E := \{e\}$ and $F := \{f\}$. Let the two input groups $X \in \{a, b\}$ each have probability $1/2$. In the coordinate order

(e, f, o_1, o_2), let the predicted and true conditional distributions be, respectively,

$$p^{(a)} = (1/4, 1/4, 2/5, 1/10), \tag{85}$$

$$p^{(b)} = (1/4, 1/4, 1/10, 2/5), \tag{86}$$

$$\eta^{(a)} = (3/8, 1/8, 2/5, 1/10), \tag{87}$$

$$\eta^{(b)} = (1/8, 3/8, 1/10, 2/5). \tag{88}$$

Thus $P = p^{(g)}$ when $X = g$.

Zero UC for the subtree indicator utilities. For a leaf subset A, define $u_A(p, y) := \mathbf{1}\{y \in A\}$, $p(A) := \sum_{y \in A} p_y$, and $\mu_{u_A}(p) = p(A)$. Both groups have $P(D) = 1/2$ and $P(E) = 1/4$. Conditioning on either scalar score therefore retains both groups with equal probability, and

$$\mathbb{P}(Y \in D \mid P(D) = 1/2) = \frac{1}{2}(\frac{3}{8} + \frac{1}{8}) + \frac{1}{2}(\frac{1}{8} + \frac{3}{8}) = \frac{1}{2},$$
$$\mathbb{P}(Y \in E \mid P(E) = 1/4) = \frac{1}{2} \cdot \frac{3}{8} + \frac{1}{2} \cdot \frac{1}{8} = \frac{1}{4}.$$

Hence $\mathbb{E}[u_A(P, Y) \mid P(A)] = P(A)$ for $A = D, E$. Substituting this identity into the UC definition gives, for every interval I,

$$\mathbb{E}\big[\mathbf{1}\{P(A) \in I\}\big(u_A(P, Y) - P(A)\big)\big]$$
$$= \mathbb{E}\big[\mathbf{1}\{P(A) \in I\}\big(\mathbb{E}[u_A(P, Y) \mid P(A)] - P(A)\big)\big] = 0.$$

Therefore $\mathrm{UC}(f; \{1\}, \{u_D, u_E\}) = 0$.

Zero class-wise calibration error. Class-wise calibration requires $\mathbb{P}(Y = k \mid P_k) = P_k$ for each leaf k. For e and f, the scalar predictions are constant at $1/4$, and

$$\mathbb{P}(Y = e \mid P_e = 1/4) = \frac{1}{2} \cdot \frac{3}{8} + \frac{1}{2} \cdot \frac{1}{8} = \frac{1}{4},$$
$$\mathbb{P}(Y = f \mid P_f = 1/4) = \frac{1}{2} \cdot \frac{1}{8} + \frac{1}{2} \cdot \frac{3}{8} = \frac{1}{4}.$$

For o_1 and o_2, each scalar prediction identifies one group, whose true probability agrees with that prediction:

$$\mathbb{P}(Y = o_1 \mid P_{o_1} = 2/5) = \eta^{(a)}_{o_1} = \frac{2}{5}, \qquad \mathbb{P}(Y = o_1 \mid P_{o_1} = 1/10) = \eta^{(b)}_{o_1} = \frac{1}{10},$$
$$\mathbb{P}(Y = o_2 \mid P_{o_2} = 1/10) = \eta^{(a)}_{o_2} = \frac{1}{10}, \qquad \mathbb{P}(Y = o_2 \mid P_{o_2} = 2/5) = \eta^{(b)}_{o_2} = \frac{2}{5}.$$

These calculations cover every possible scalar prediction for all four leaves, so class-wise calibration holds exactly.

Nonzero conditional branch-probability errors. The conditional branch agreement discussed in Section 2.2 would require $\mathbb{P}(Y \in E \mid Y \in D, P) = P(E)/P(D)$. The right-hand side is the predicted probability of taking the $D \to E$ branch. In both groups it equals $\frac{p^{(a)}(E)}{p^{(a)}(D)} = \frac{p^{(b)}(E)}{p^{(b)}(D)} = \frac{1/4}{1/2} = \frac{1}{2}$. However, $p^{(a)} \neq p^{(b)}$, so the full prediction vector identifies the group. The true conditional branch probabilities are

$$\mathbb{P}(Y \in E \mid Y \in D, P = p^{(a)}) = \frac{\eta^{(a)}(E)}{\eta^{(a)}(D)} = \frac{3/8}{3/8 + 1/8} = \frac{3}{4},$$
$$\mathbb{P}(Y \in E \mid Y \in D, P = p^{(b)}) = \frac{\eta^{(b)}(E)}{\eta^{(b)}(D)} = \frac{1/8}{1/8 + 3/8} = \frac{1}{4}.$$

Thus the errors are

$$\mathbb{P}(Y \in E \mid Y \in D, P = p^{(a)}) - \frac{p^{(a)}(E)}{p^{(a)}(D)} = \frac{3}{4} - \frac{1}{2} = \frac{1}{4},$$
$$\mathbb{P}(Y \in E \mid Y \in D, P = p^{(b)}) - \frac{p^{(b)}(E)}{p^{(b)}(D)} = \frac{1}{4} - \frac{1}{2} = -\frac{1}{4}.$$

The $D \to E$ probability is underestimated by $1/4$ in group a and overestimated by $1/4$ in group b.

The reason these errors do not appear in the indicator-utility UC is also explicit. The conditional mean utility residuals are

$$\mathbb{E}[u_E(P,Y) \mid P = p^{(a)}] - \mu_{u_E}(p^{(a)}) = \frac{3}{8} - \frac{1}{4} = \frac{1}{8},$$
$$\mathbb{E}[u_E(P,Y) \mid P = p^{(b)}] - \mu_{u_E}(p^{(b)}) = \frac{1}{8} - \frac{1}{4} = -\frac{1}{8}.$$

Both groups have predicted mean utility $1/4$, so a predicted-utility interval cannot separate them, and their residuals cancel: $\frac{1}{2} \cdot \frac{1}{8} + \frac{1}{2}(-\frac{1}{8}) = 0$. The same cancellation occurs in class-wise calibration of e, since both groups have $P_e = 1/4$. Consequently, zero indicator-utility UC and zero class-wise calibration error do not guarantee agreement of branch probabilities conditional on the full prediction vector. In this example the conditional branch-probability error has absolute value $1/4$ in each group.

D.3 Branch Errors versus Utility Impact

Section 2.2 distinguishes a branch-probability error from its effect on the specified utility. We verify this distinction by substituting into the three contributions D_{top}, D_L, D_R defined in Eq. (5).

Use the four-leaf tree of that section, with y_1, y_2 below v_L and y_3, y_4 below v_R. Let the predictor be constant at $p = (1/4, 1/4, 1/4, 1/4)$, let the true conditional distribution be $\eta = (1/20, 9/20, 1/5, 3/10)$, and take the target utility to be $u(p, y) := \mathbf{1}\{y = y_4\}$, $(u_1, u_2, u_3, u_4) = (0, 0, 0, 1)$.

Branch-probability errors. The predicted right-branch probabilities are $q_{0,R} = p_3 + p_4 = \frac{1}{2}$, $q_{L,R} = \frac{p_2}{p_1+p_2} = \frac{1}{2}$, $q_{R,R} = \frac{p_4}{p_3+p_4} = \frac{1}{2}$. The corresponding true probabilities are

$$q^\star_{0,R} = \eta_3 + \eta_4 = \frac{1}{5} + \frac{3}{10} = \frac{1}{2}, \qquad q^\star_{L,R} = \frac{\eta_2}{\eta_1 + \eta_2} = \frac{9/20}{1/20 + 9/20} = \frac{9}{10},$$
$$q^\star_{R,R} = \frac{\eta_4}{\eta_3 + \eta_4} = \frac{3/10}{1/5 + 3/10} = \frac{3}{5}.$$

Hence $|q^\star_{0,R} - q_{0,R}| = 0$, $|q^\star_{L,R} - q_{L,R}| = \frac{2}{5}$, $|q^\star_{R,R} - q_{R,R}| = \frac{1}{10}$. The left node has the largest branch-probability error.

Contributions to the target-utility error. The predicted subtree mean utilities from Section 2.2 are $\bar{u}_{0,L} = q_{L,L}u_1 + q_{L,R}u_2 = 0$ and $\bar{u}_{0,R} = q_{R,L}u_3 + q_{R,R}u_4 = 1/2$. Direct substitution into Eq. (5) gives

$$D_{\text{top}}(p) = (q^\star_{0,R} - q_{0,R})(\bar{u}_{0,R} - \bar{u}_{0,L}) = (\frac{1}{2} - \frac{1}{2})(\frac{1}{2} - 0) = 0,$$
$$D_L(p) = q^\star_{0,L}(q^\star_{L,R} - q_{L,R})(u_2 - u_1) = (1 - \frac{1}{2})(\frac{9}{10} - \frac{1}{2})(0 - 0) = 0,$$
$$D_R(p) = q^\star_{0,R}(q^\star_{R,R} - q_{R,R})(u_4 - u_3) = \frac{1}{2}(\frac{3}{5} - \frac{1}{2})(1 - 0) = \frac{1}{20}.$$

At the left node, the branch-probability error is $2/5$, but the utility difference $u_2 - u_1$ multiplying it is zero. At the right node, the smaller branch-probability error $1/10$ is multiplied by the nonzero utility difference $u_4 - u_3 = 1$, giving contribution $1/20$. Indeed,

$$D^{\text{full}}(p) = \sum_{i=1}^{4} \eta_i u_i - \sum_{i=1}^{4} p_i u_i = \frac{3}{10} - \frac{1}{4} = \frac{1}{20} = D_{\text{top}}(p) + D_L(p) + D_R(p).$$

Thus the target-utility error in this example comes from the right node, whereas ranking nodes only by branch-probability error places the left node first, despite its zero utility contribution. Branch-probability error alone therefore does not identify the source of the specified target-utility error.

D.4 Node-Specific Branch Cells Can Audit Different Populations

Section 2.2 seeks to compare the sources of utility error within the same population selected by predicted mean utility. This example shows that selecting a population by each node's own branch score need not preserve that common population.

Use the same four-leaf tree and two input groups a, b, each with probability $1/2$. Let the predicted distributions be $p^{(a)} = (1/2, 1/10, 3/20, 1/4)$, $p^{(b)} = (1/5, 3/10, 1/4, 1/4)$, and the true conditional distributions be $\eta^{(a)} = (5/12, 1/12, 3/16, 5/16)$, $\eta^{(b)} = (1/5, 3/10, 3/10, 1/5)$. Again take $u(p, y) := \mathbf{1}\{y = y_4\}$.

The same predicted-utility cell. By the definition of predicted mean utility, $\mu_u(p^{(a)}) = p_4^{(a)} = \frac{1}{4}$, $\mu_u(p^{(b)}) = p_4^{(b)} = \frac{1}{4}$. Both groups therefore belong to the same cell $\{\mu_u(P) = 1/4\}$.

Different sources of utility error in the two groups. The root and right-node predicted branch probabilities are

$$q_{0,R}(p^{(a)}) = \frac{3}{20} + \frac{1}{4} = \frac{2}{5}, \qquad q_{R,R}(p^{(a)}) = \frac{1/4}{3/20 + 1/4} = \frac{5}{8},$$
$$q_{0,R}(p^{(b)}) = \frac{1}{4} + \frac{1}{4} = \frac{1}{2}, \qquad q_{R,R}(p^{(b)}) = \frac{1/4}{1/4 + 1/4} = \frac{1}{2}.$$

The corresponding true probabilities are

$$q^\star_{0,R}(p^{(a)}) = \frac{3}{16} + \frac{5}{16} = \frac{1}{2}, \qquad q^\star_{R,R}(p^{(a)}) = \frac{5/16}{3/16 + 5/16} = \frac{5}{8},$$
$$q^\star_{0,R}(p^{(b)}) = \frac{3}{10} + \frac{1}{5} = \frac{1}{2}, \qquad q^\star_{R,R}(p^{(b)}) = \frac{1/5}{3/10 + 1/5} = \frac{2}{5}.$$

For this utility, $\bar{u}_{0,L} = 0$, $\bar{u}_{0,R} = q_{R,R}$, $u_2 - u_1 = 0$, and $u_4 - u_3 = 1$. Substituting into Eq. (5) yields

$$D_{\text{top}}(p^{(a)}) = (\frac{1}{2} - \frac{2}{5})(\frac{5}{8} - 0) = \frac{1}{16}, \qquad D_R(p^{(a)}) = \frac{1}{2}(\frac{5}{8} - \frac{5}{8})(1 - 0) = 0,$$
$$D_{\text{top}}(p^{(b)}) = (\frac{1}{2} - \frac{1}{2})(\frac{1}{2} - 0) = 0, \qquad D_R(p^{(b)}) = \frac{1}{2}(\frac{2}{5} - \frac{1}{2})(1 - 0) = -\frac{1}{20}.$$

The left-node contribution is zero in both groups because $u_2 - u_1 = 0$. Thus, among observations with the same predicted mean utility $1/4$, group a has a root contribution $1/16$, whereas group b has a right-node contribution $-1/20$.

Branch-score cells do not preserve the common population. For the root score, take $I_0 = [3/8, 7/16]$. Then $q_{0,R}(p^{(a)}) = \frac{2}{5} \in I_0$, $q_{0,R}(p^{(b)}) = \frac{1}{2} \notin I_0$, so this cell selects only group a. For the right-node score, take $I_R = [7/16, 9/16]$. Then $q_{R,R}(p^{(a)}) = \frac{5}{8} \notin I_R$, $q_{R,R}(p^{(b)}) = \frac{1}{2} \in I_R$, so this cell selects only group b. Equivalently, $\mathbf{1}\{q_{0,R}(P) \in I_0\} = \mathbf{1}\{X = a\}$, $\mathbf{1}\{q_{R,R}(P) \in I_R\} = \mathbf{1}\{X = b\}$. The two cells select disjoint input populations. Their results therefore do not compare the root and right-node contributions within the same population.

By contrast, retaining the common predicted-utility interval $I = [1/4, 1/4]$ from Section 2.2 gives

$$\mathbb{E}[\mathbf{1}\{\mu_u(P) \in I\} D_{\text{top}}(P)] = \frac{1}{2} \cdot \frac{1}{16} + \frac{1}{2} \cdot 0 = \frac{1}{32},$$
$$\mathbb{E}[\mathbf{1}\{\mu_u(P) \in I\} D_L(P)] = 0,$$
$$\mathbb{E}[\mathbf{1}\{\mu_u(P) \in I\} D_R(P)] = \frac{1}{2} \cdot 0 + \frac{1}{2}(-\frac{1}{20}) = -\frac{1}{40}.$$

All three expectations concern the same population containing both groups. Comparing the hierarchical sources of the target-utility error requires retaining this common predicted-utility cell rather than changing the selected population from node to node.

Auditing every branch also includes utility-irrelevant fluctuations. At the left node, the predicted and true branch probabilities are

$$q_{L,R}(p^{(a)}) = \frac{1/10}{1/2+1/10} = \frac{1}{6}, \qquad q^\star_{L,R}(p^{(a)}) = \frac{1/12}{5/12+1/12} = \frac{1}{6},$$
$$q_{L,R}(p^{(b)}) = \frac{3/10}{1/5+3/10} = \frac{3}{5}, \qquad q^\star_{L,R}(p^{(b)}) = \frac{3/10}{1/5+3/10} = \frac{3}{5}.$$

Thus the population branch-probability error is zero in each group. Nevertheless, estimating these probabilities from a finite sample introduces fluctuations in the observed leaf frequencies. For example, conditional on group a and on reaching the left subtree, the difference from the predicted probability $1/6$ is

$$\mathbf{1}\{Y = y_2\} - \frac{1}{6} = \begin{cases} -\frac{1}{6}, & Y = y_1, \\ \frac{5}{6}, & Y = y_2. \end{cases}$$

Its population mean is $\frac{5}{6}(-\frac{1}{6}) + \frac{1}{6}\cdot\frac{5}{6} = 0$, but its finite-sample mean need not be zero. A full branch audit therefore includes fluctuating empirical discrepancies in the cells at scores $1/6$ and $3/5$.

For the specified target utility, however, the left-node term in Eq. (5) is multiplied by $u_2 - u_1 = 0$, so $D_L(p) = q^\star_{0,L}(q^\star_{L,R} - q_{L,R})\underbrace{(u_2 - u_1)}_{=0} = 0$. This remains zero regardless of the branch-probability error or its empirical estimate. The fluctuating allocation between y_1 and y_2 does not affect this utility. Auditing every branch would include these utility-irrelevant fluctuations. To locate the target-utility error in this example, it suffices to compare the root and right-node contributions, which can be nonzero, on the same predicted-utility cell.

E Proofs for HUC-Boost

This appendix proves the loss decrease and finite termination stated in Theorem 3.5. Appendix E.1 prepares the softmax branch probabilities, the induced leaf distribution, and log loss. Appendix E.2 then proves the theorem in four steps. We then compare branch-local updates with ordinary leaf Brier loss and prove the auxiliary path-product results.

E.1 Softmax Branches, Leaf Distribution, and Log Loss

Before the main proof, we verify that node-wise softmax branch probabilities define a positive leaf distribution, that the original conditional branch probabilities are recovered from this leaf distribution, and that the population log loss used as the progress potential is well defined.

For a predictor $g : \mathcal{X} \to \Delta^{K-1}$, define the population log loss by

$$\mathcal{L}_{\log}(g) := \mathbb{E}[-\log g(X)_Y]. \tag{89}$$

Assign logits $\boldsymbol{z}_v(x) = (z_{v,1}(x), \dots, z_{v,d_v}(x))^\top$ to each internal node v, and set

$$q^{\boldsymbol{z}}_{v,j}(x) := \frac{\exp(z_{v,j}(x))}{\sum_{k=1}^{d_v} \exp(z_{v,k}(x))}, \qquad j = 1, \dots, d_v. \tag{90}$$

For every leaf y, use the path sequence in Eq. (22) to define

$$f_{\boldsymbol{z}}(x)_y := \prod_{r=0}^{L_y-1} q^{\boldsymbol{z}}_{w_r(y),j_r(y)}(x). \tag{91}$$

Lemma E.6 shows that this path product defines a positive probability distribution over the leaves for every input. Lemma E.7 further gives, for every internal node v and child v_j,

$$\pi_{v_j}(f_{\boldsymbol{z}}(x)) = \pi_v(f_{\boldsymbol{z}}(x)) q^{\boldsymbol{z}}_{v,j}(x). \tag{92}$$

Hence the conditional branch probability recovered from the leaf distribution by Eq. (28) equals the original softmax branch probability. Appendix E.4 proves both lemmas.

E.2 Proof of Theorem 3.5

The proof proceeds in four steps. Step 1 derives how a one-node logit update changes branch and leaf probabilities and identifies the probability structure it preserves. Step 2 evaluates the resulting change in ordinary leaf log loss. Step 3 computes the first and second derivatives needed for the Taylor expansion in Step 4. Step 4 establishes the one-step loss decrease and finite termination.

Step 1: probability changes under a one-node logit update. At iteration t, let (c_t, u_t, I_t, v_t) be the selected candidate, and write $v := v_t$ and $d := d_v$. Define the update direction at the selected node by

$$\boldsymbol{h}_t(x) := c_t(x)\mathbf{1}\{\mu_{u_t}(f_t(x)) \in I_t\}\boldsymbol{\Delta}_{u_t,v}(f_t(x)). \tag{93}$$

For any $\alpha \in \mathbb{R}$, set

$$\boldsymbol{z}_{t,v}^{(\alpha)}(x) := \boldsymbol{z}_{t,v}(x) + \alpha \boldsymbol{h}_t(x), \qquad \boldsymbol{z}_{t,w}^{(\alpha)}(x) := \boldsymbol{z}_{t,w}(x) \quad (w \neq v), \tag{94}$$

and denote the corresponding branch probabilities and leaf distribution by $\boldsymbol{q}_{t,w}^{(\alpha)}$ and $f_t^{(\alpha)}$.

Proposition E.1 (Branch and leaf probabilities after a one-node update)**.** *Let*

$$Z_t^{(\alpha)}(x) := \sum_{k=1}^{d} q_{t,v,k}(x) e^{\alpha h_{t,k}(x)}. \tag{95}$$

At the selected node,

$$q_{t,v,j}^{(\alpha)}(x) = \frac{q_{t,v,j}(x) e^{\alpha h_{t,j}(x)}}{Z_t^{(\alpha)}(x)}, \qquad j = 1, \dots, d, \tag{96}$$

whereas the branch probabilities at every $w \neq v$ remain unchanged. Moreover, for every leaf y,

$$f_t^{(\alpha)}(x)_y = \begin{cases} f_t(x)_y, & y \notin \mathcal{Y}_v, \\ f_t(x)_y \frac{e^{\alpha h_{t,j}(x)}}{Z_t^{(\alpha)}(x)}, & y \in \mathcal{Y}_{v_j}. \end{cases} \tag{97}$$

Consequently, the update preserves every leaf probability outside the selected subtree, the reach probability of the selected node, and the conditional leaf distribution within each child subtree.

Proof. By the softmax definition,

$$q_{t,v,j}^{(\alpha)}(x) = \frac{e^{z_{t,v,j}(x) + \alpha h_{t,j}(x)}}{\sum_{k=1}^{d} e^{z_{t,v,k}(x) + \alpha h_{t,k}(x)}} = \frac{q_{t,v,j}(x) e^{\alpha h_{t,j}(x)}}{\sum_{k=1}^{d} q_{t,v,k}(x) e^{\alpha h_{t,k}(x)}}.$$

This proves Eq. (96). If the path to y does not pass through v, none of its path-product factors changes. If $y \in \mathcal{Y}_{v_j}$, the only changed factor is $q_{t,v,j}$, yielding Eq. (97).

Summing Eq. (97) over $\mathcal{Y}_{v_j}$ gives

$$\pi_{v_j}(f_t^{(\alpha)}(x)) = \pi_v(f_t(x)) q_{t,v,j}^{(\alpha)}(x).$$

Summing further over j yields $\pi_v(f_t^{(\alpha)}(x)) = \pi_v(f_t(x))$. Hence, for $y \in \mathcal{Y}_{v_j}$,

$$f_t^{(\alpha)}(x)_y / \pi_{v_j}(f_t^{(\alpha)}(x)) = f_t(x)_y / \pi_{v_j}(f_t(x)),$$

which proves preservation of the conditional leaf distribution within every child subtree. □

Step 2: change in the potential under a one-node update. Recall the subtree-membership indicators $J_v(y) = \mathbf{1}\{y \in \mathcal{Y}_v\}$ and $J_{v,j}(y) = \mathbf{1}\{y \in \mathcal{Y}_{v_j}\}$, where $\mathcal{Y}_v$ and $\mathcal{Y}_{v_j}$ are the descendant-leaf sets of v and its child v_j. We now evaluate how the one-node logit update changes the population log loss defined in Appendix E.1.

Lemma E.2 (Exact change in log loss under a one-node update)**.** *For every* (x, y)*,*

$$-\log f_t^{(\alpha)}(x)_y + \log f_t(x)_y = J_v(y) \log Z_t^{(\alpha)}(x) - \alpha \sum_{j=1}^{d} J_{v,j}(y) h_{t,j}(x). \tag{98}$$

Consequently,

$$\mathcal{L}_{\log}(f_t^{(\alpha)}) - \mathcal{L}_{\log}(f_t) = \mathbb{E}\left[J_v(Y) \log Z_t^{(\alpha)}(X) - \alpha \sum_{j=1}^{d} J_{v,j}(Y) h_{t,j}(X) \right]. \tag{99}$$

Proof. Proposition E.1 gives, for every leaf y,

$$\frac{f_t^{(\alpha)}(x)_y}{f_t(x)_y} = \exp\left(\alpha \sum_{j=1}^{d} J_{v,j}(y) h_{t,j}(x) - J_v(y) \log Z_t^{(\alpha)}(x) \right).$$

If $y \notin \mathcal{Y}_v$, all indicators on the right are zero. If $y \in \mathcal{Y}_{v_j}$, the right-hand side equals $e^{\alpha h_{t,j}(x)} / Z_t^{(\alpha)}(x)$. Negative logarithms give Eq. (98), and expectations give Eq. (99). □

Step 3: first and second derivatives needed for the one-step decrease. For the Taylor expansion in Step 4, define $\phi_t(\alpha) := \mathcal{L}_{\log}(f_t^{(\alpha)})$ and evaluate its first derivative at $\alpha = 0$ and its second derivative at arbitrary α. Set

$$\Gamma_t := \Gamma_{c_t, u_t, I_t, v_t}(f_t) = \mathbb{E}\left[\langle \boldsymbol{h}_t(X), \boldsymbol{R}_v(f_t(X), Y) \rangle \right], \tag{100}$$

$$\Lambda_t := \frac{1}{4} \mathbb{E}\left[J_v(Y) \left\{ \max_j h_{t,j}(X) - \min_j h_{t,j}(X) \right\}^2 \right]. \tag{101}$$

Lemma E.3 (Directional derivative and curvature along a one-node update)**.** *Assume* $\mathcal{L}_{\log}(f_t) < \infty$*. Then* $\phi_t(\alpha) < \infty$ *for every finite* $\alpha \in \mathbb{R}$*, and*

$$\phi_t'(0) = -\Gamma_t, \tag{102}$$

$$\phi_t''(\alpha) = \mathbb{E}\left[J_v(Y) \operatorname{Var}_{J \sim \boldsymbol{q}_{t,v}^{(\alpha)}(X)}(h_{t,J}(X)) \right] \le \Lambda_t \le 1. \tag{103}$$

Proof. From Eq. (99),

$$\frac{d}{d\alpha} \log Z_t^{(\alpha)}(x) = \sum_{j=1}^{d} q_{t,v,j}^{(\alpha)}(x) h_{t,j}(x),$$

and therefore

$$\phi_t'(\alpha) = -\mathbb{E}\left[\sum_{j=1}^{d} h_{t,j}(X) \left\{ J_{v,j}(Y) - J_v(Y) q_{t,v,j}^{(\alpha)}(X) \right\} \right].$$

Setting $\alpha = 0$ gives $\phi_t'(0) = -\Gamma_t$. Moreover,

$$\frac{d^2}{d\alpha^2} \log Z_t^{(\alpha)}(x) = \operatorname{Var}_{J \sim \boldsymbol{q}_{t,v}^{(\alpha)}(x)}(h_{t,J}(x)),$$

which proves the equality in Eq. (103).

A random variable supported on $[a, b]$ has variance at most $(b-a)^2/4$. Applying this pointwise in x gives $\phi_t''(\alpha) \le \Lambda_t$. Since $\max_j h_{t,j}(x) - \min_j h_{t,j}(x) \le 2$, we also have $\Lambda_t \le 1$. The absolute value of Eq. (98) is at most $2|\alpha|$, and the first and second derivatives are uniformly bounded; hence differentiation and expectation may be interchanged. □

Step 4: one-step loss decrease and finite termination.

Lemma E.4 (Positive curvature for a nonzero audit moment)**.** *If $|\Gamma_t| > 0$, then $\Lambda_t > 0$.*

Proof. We prove the contrapositive. If $\Lambda_t = 0$, then on every point with $J_v(Y) = 1$, all components of $\boldsymbol{h}_t(X)$ are equal. Since $\sum_j R_{v,j}(f_t(X), Y) = 0$, $\langle \boldsymbol{h}_t(X), \boldsymbol{R}_v(f_t(X), Y)\rangle = 0$. When $J_v(Y) = 0$, $\boldsymbol{R}_v(f_t(X), Y) = \mathbf{0}$. Therefore $\Gamma_t = 0$. □

Proof of Theorem 3.5. Lemma E.4 gives $\Lambda_t > 0$ whenever an update is made. Lemma E.3 and the integral form of Taylor's theorem imply, for every $\alpha \in \mathbb{R}$,

$$\phi_t(\alpha) \le \phi_t(0) - \alpha\Gamma_t + \frac{\alpha^2}{2}\Lambda_t.$$

Substituting $\alpha = \alpha_t = \Gamma_t/\Lambda_t$ gives the one-step loss decrease used in Theorem 3.5. Proposition E.1 ensures that all branch and leaf probabilities remain positive after every finite number of updates.

Before termination, $|\Gamma_t| > \varepsilon$ and $\Lambda_t \le 1$, so each update decreases log loss by more than $\varepsilon^2/2$. If m updates are made, nonnegativity of log loss gives

$$\mathcal{L}_{\log}(f_0) \ge \sum_{t=0}^{m-1} \{\mathcal{L}_{\log}(f_t) - \mathcal{L}_{\log}(f_{t+1})\} > m\frac{\varepsilon^2}{2}.$$

Thus the update bound in Theorem 3.5 follows. At termination, $|\Gamma_{c,u,I,v}(f_T)| \le \varepsilon$ for every candidate, and hence $\mathrm{HUC}(f_T; \mathcal{C}, \mathcal{U}) \le \varepsilon$. □

Fixed-step empirical update. On a fixed sample $S = ((X_i, Y_i))_{i=1}^n$, define

$$\widehat{\mathcal{L}}_{\log,S}(g) := -\frac{1}{n}\sum_{i=1}^{n} \log g(X_i)_{Y_i}. \tag{104}$$

Given the current predictor f and a candidate (c, u, I, v), set $\boldsymbol{h}(x) := c(x)\mathbf{1}\{\mu_u(f(x)) \in I\}\boldsymbol{\Delta}_{u,v}(f(x))$, and denote the corresponding empirical audit moment by $\widehat{\Gamma}$.

Corollary E.5 (Empirical loss decrease under a signed fixed-step update)**.** *For any $\alpha > 0$ and $\sigma \in \{-1, 1\}$, let f^+ be the predictor obtained by updating the logits at the selected node as $\boldsymbol{z}_v^+ = \boldsymbol{z}_v + \sigma\alpha\boldsymbol{h}$. Then*

$$\widehat{\mathcal{L}}_{\log,S}(f^+) - \widehat{\mathcal{L}}_{\log,S}(f) \le -\alpha\sigma\widehat{\Gamma} + \frac{\alpha^2}{2}. \tag{105}$$

In particular, if $\sigma = \mathrm{sgn}(\widehat{\Gamma})$, $|\widehat{\Gamma}| > 3\varepsilon/4$, and $\alpha = \varepsilon/2$, then the empirical log loss decreases by more than $\varepsilon^2/4$.

Proof. Apply the Taylor bound of Lemma E.3 to the empirical average and use the curvature bound 1. □

E.3 Ordinary Leaf Brier Loss Is Not Branch-Local

Define the ordinary multiclass Brier loss by

$$\mathcal{B}_{\mathrm{leaf}}(p, y) := \|e_y - p\|_2^2 \tag{106}$$

For UC-Boost, which updates the full leaf-probability vector, the first-order term of this loss aligns with the leaf residual $e_y - p$. A branch-local HUC-Boost update preserves the parent reach probability and the conditional leaf distributions within child subtrees, so its direction in leaf space differs.

Fix a positive prediction vector p, an internal node v, and its children $v_1, \ldots, v_{d_v}$. Define the conditional leaf distribution within each child subtree, embedded in the full leaf space, and its squared norm by

$$[\boldsymbol{r}_{v,j}(p)]_z := \frac{p_z}{\pi_{v_j}(p)}\mathbf{1}\{z \in \mathcal{Y}_{v_j}\}, \qquad C_{v,j}(p) := \|\boldsymbol{r}_{v,j}(p)\|_2^2 \tag{107}$$

Hold all branches other than v fixed, and let $\boldsymbol{q}'_v = \boldsymbol{q}_v(p)+\boldsymbol{\delta}$, where $\sum_j \delta_j = 0$ and every component of $\boldsymbol{q}'_v$ is positive. The same path-product calculation as in Step 1 gives

$$p' - p = \pi_v(p) \sum_{j=1}^{d_v} \delta_j \boldsymbol{r}_{v,j}(p). \tag{108}$$

The supports of the $\boldsymbol{r}_{v,j}$ are disjoint, and $\langle \boldsymbol{r}_{v,j}(p), p\rangle = \pi_v(p) q_{v,j}(p) C_{v,j}(p)$. Hence

$$\begin{aligned}
&\mathcal{B}_{\mathrm{leaf}}(p', y) - \mathcal{B}_{\mathrm{leaf}}(p, y) \\
&\quad = -2\langle e_y - p, p' - p\rangle + \|p' - p\|_2^2 \\
&\quad = -2\pi_v(p) \sum_{j=1}^{d_v} \delta_j \left\{ [\boldsymbol{r}_{v,j}(p)]_y - \pi_v(p) q_{v,j}(p) C_{v,j}(p) \right\} \\
&\qquad + \pi_v(p)^2 \sum_{j=1}^{d_v} \delta_j^2 C_{v,j}(p).
\end{aligned} \tag{109}$$

The first-order term in Eq. (109) does not generally align with the local HUC residual. To see this, consider an observed label outside the selected subtree, $y \notin \mathcal{Y}_v$. Then $J_v(y) = J_{v,j}(y) = 0$ and $\boldsymbol{R}_v(p, y) = \mathbf{0}$. Since $[\boldsymbol{r}_{v,j}(p)]_y = 0$ as well, Eq. (109) becomes

$$\mathcal{B}_{\mathrm{leaf}}(p', y) - \mathcal{B}_{\mathrm{leaf}}(p, y) = 2\pi_v(p)^2 \sum_j \delta_j q_{v,j}(p) C_{v,j}(p) + \pi_v(p)^2 \sum_j \delta_j^2 C_{v,j}(p).$$

Although $\sum_j \delta_j = 0$, the coefficients $q_{v,j}(p)C_{v,j}(p)$ generally differ across children, so the linear term need not vanish. For example, setting $\delta_j = t$, $\delta_k = -t$, and all other components to zero gives the linear term

$$2\pi_v(p)^2 t \{q_{v,j}(p)C_{v,j}(p) - q_{v,k}(p)C_{v,k}(p)\}.$$

If the expression in braces is positive, the leaf Brier loss increases for sufficiently small $t > 0$. Thus the first-order change in leaf Brier loss need not vanish even when the local HUC residual is zero, and a decrease guarantee cannot be obtained by directly identifying the HUC moment with this first-order term.

Lifting to leaf space versus branch locality. Lift the HUC node contribution to leaf space by setting

$$g_{u,v}(p)_z := \langle \boldsymbol{\Delta}_{u,v}(p), \boldsymbol{R}_v(p, z)\rangle \tag{110}$$

Since $\sum_z p_z \boldsymbol{R}_v(p, z) = \mathbf{0}$,

$$\langle \boldsymbol{g}_{u,v}(p), e_Y - p\rangle = \langle \boldsymbol{\Delta}_{u,v}(p), \boldsymbol{R}_v(p, Y)\rangle .$$

Thus a projected update on the full leaf simplex can use the HUC moment as the first-order Brier term. However, for $z \in \mathcal{Y}_{v_j}$, $g_{u,v}(p)_z = \mu_{u,v_j}(p) - \mu_{u,v}(p)$, so the same amount is added to all leaves in a child subtree. Unlike Eq. (108), this change is not proportional to conditional leaf probabilities and generally changes relative probabilities within a child subtree. Simplex projection can also alter leaves outside the selected subtree, so the update is not branch-local.

The logit update in Appendix E.2 is paired with ordinary global leaf log loss.

E.4 Auxiliary Results for Softmax Path Products

This subsection proves the normalization of the path product and its consistency with the branch probabilities used in Appendix E.1.

Lemma E.6 (Path products define a probability distribution on the leaves)**.** *For every input x,*

$$f_{\boldsymbol{z}}(x)_y > 0 \quad (y \in \mathcal{Y}), \qquad \sum_{y\in\mathcal{Y}} f_{\boldsymbol{z}}(x)_y = 1. \tag{111}$$

Therefore $f_{\boldsymbol{z}} : \mathcal{X} \to \Delta^{K-1}$.

Proof. Fix x. Each softmax vector in Eq. (90) has positive coordinates summing to one. Starting at the root, whenever an internal node v is reached, choose its child v_j with probability $q^{\boldsymbol{z}}_{v,j}(x)$ and continue until a leaf is reached. This specifies a valid probability distribution for the next branch at every step. Since the tree is finite and each move goes to a child, the procedure terminates at exactly one leaf.

For each leaf y, the multiplication rule for conditional probabilities along its unique root-to-leaf path gives its probability of being reached as $\prod_{r=0}^{L_y-1} q^{\boldsymbol{z}}_{w_r(y),j_r(y)}(x) = f_{\boldsymbol{z}}(x)_y > 0$. The events of terminating at the respective leaves are disjoint and exhaustive, so $\sum_{y\in\mathcal{Y}} f_{\boldsymbol{z}}(x)_y = 1$. □

Lemma E.7 (Consistency of the generated leaf distribution and the node branch probabilities)**.** *Fix an input x and set $p := f_{\boldsymbol{z}}(x)$. For every internal node v and child v_j,*

$$\pi_{v_j}(p) = \pi_v(p) q^{\boldsymbol{z}}_{v,j}(x). \tag{112}$$

Thus the predicted conditional branch probabilities computed from p by Eq. (28) *coincide with the original softmax probabilities $q^{\boldsymbol{z}}_{v,j}(x)$.*

Proof. Use the root-to-leaf procedure in the proof of Lemma E.6, whose terminal-leaf distribution is p. Terminating in $\mathcal{Y}_v$ is equivalent to reaching v, so $\pi_v(p) > 0$ is the probability of reaching v. Conditional on reaching v, the next step selects v_j with probability $q^{\boldsymbol{z}}_{v,j}(x)$. The terminal leaf lies in $\mathcal{Y}_{v_j}$ if and only if this child is selected. Since $\mathcal{Y}_{v_j} \subseteq \mathcal{Y}_v$, the definition of conditional probability gives $\frac{\pi_{v_j}(p)}{\pi_v(p)} = q^{\boldsymbol{z}}_{v,j}(x)$. Multiplying by $\pi_v(p)$ proves Eq. (112), and the ratio is exactly the recovered branch probability in Eq. (28). □

F Finite-Sample Evaluation and Exact Auditing

Section 3.1 ends with the finite-sample result. This appendix proves Theorem 3.4, tracks the numerical constants in the uniform deviation bound, and explains exact empirical maximization over all intervals on a finite sample. The proof follows the same outline as the finite-sample analysis of UC. For a fixed triple (c, u, v), we reduce intervals to half-lines, evaluate score-ordered Rademacher partial sums, and then apply symmetrization, concentration, and a union bound over the N_{cand} triples. The predictor is fixed independently of the audit sample; sample-adaptive prediction is treated in Appendix H.

F.1 Proof of Theorem 3.4

Proof. We divide the proof into four steps. The first three fix one triple (c, u, v) and vary only the interval I. The final step treats all triples simultaneously.

Step 1: bounding the contribution of one observation. Fix $c \in \mathcal{C}$, $u \in \mathcal{U}$, and $v \in \mathcal{V}_{\mathcal{T}}(u)$, and set

$$A(X, Y) := c(X) \left\langle \boldsymbol{\Delta}_{u,v}(P), \boldsymbol{R}_v(P, Y) \right\rangle, \qquad s(X) := \mu_u(P). \tag{113}$$

If $Y \notin \mathcal{Y}_v$, Lemma C.2 gives $A(X, Y) = 0$. If $Y \in \mathcal{Y}_{v_j}$, the same lemma gives

$$A(X, Y) = c(X)\{\mu_{u,v_j}(P) - \mu_{u,v}(P)\}. \tag{114}$$

Subtree mean utilities are convex combinations of values in $[-1, 1]$, so both $\mu_{u,v_j}(P)$ and $\mu_{u,v}(P)$ belong to $[-1, 1]$. Hence $|A(X, Y)| \le |c(X)|\,|\mu_{u,v_j}(P) - \mu_{u,v}(P)| \le 2$. Thus, in all cases,

$$|A(X, Y)| \le 2. \tag{115}$$

For $I \in \mathcal{I}$, define

$$h_I(x, y) := A(x, y)\mathbf{1}\{s(x) \in I\}. \tag{116}$$

The population and empirical audit moments are

$$\Gamma_{c,u,I,v}(f) = \mathbb{E}[h_I(X, Y)], \qquad \widehat{\Gamma}_{c,u,I,v}(f) := \frac{1}{n}\sum_{i=1}^{n} h_I(X_i, Y_i). \tag{117}$$

Step 2: reducing intervals to half-lines. Let $\mathcal{H} := \{\varnothing, \mathbb{R}\} \cup \{(-\infty, t), (-\infty, t] : t \in \mathbb{R}\}$ be the class of left half-lines. Every real interval I can be represented, for suitable $H_1, H_0 \in \mathcal{H}$, as

$$\mathbf{1}\{s \in I\} = \mathbf{1}\{s \in H_1\} - \mathbf{1}\{s \in H_0\}. \tag{118}$$

For example, if $I = (a, b]$, take $H_1 = (-\infty, b]$ and $H_0 = (-\infty, a]$. Other endpoint conventions are handled by choosing $<$ or $\leq$, and unbounded intervals and the empty set are covered using $\mathbb{R}$ or $\varnothing$.

Fix the sample and write $A_i := A(X_i, Y_i)$ and $s_i := s(X_i)$. Introduce independent Rademacher signs $\xi_1, \ldots, \xi_n$. By Eq. (118) and the triangle inequality,

$$\begin{aligned}\sup_{I \in \mathcal{I}} \left| \frac{1}{n} \sum_{i=1}^{n} \xi_i A_i \mathbf{1}\{s_i \in I\} \right| \\ \leq 2 \sup_{H \in \mathcal{H}} \left| \frac{1}{n} \sum_{i=1}^{n} \xi_i A_i \mathbf{1}\{s_i \in H\} \right|.\end{aligned} \tag{119}$$

Fix a permutation ordering the scores nondecreasingly, and write the reordered values as $s_{(1)} \leq \cdots \leq s_{(n)}$, with corresponding $A_{(i)}$ and $\xi_{(i)}$. A half-line never splits a tied score group, and the selected indices form a prefix $\{1, \ldots, k\}$. Therefore

$$\sup_{H \in \mathcal{H}} \left| \sum_{i=1}^{n} \xi_i A_i \mathbf{1}\{s_i \in H\} \right| \leq \max_{0 \leq k \leq n} \left| \sum_{i=1}^{k} \xi_{(i)} A_{(i)} \right|. \tag{120}$$

Let $Z_0^\xi := 0$, $Z_k^\xi := \sum_{i=1}^{k} \xi_{(i)} A_{(i)}$, and $\mathcal{F}_k^\xi := \sigma(\xi_{(1)}, \ldots, \xi_{(k)})$. Conditional on the sample, $A_{(1)}, \ldots, A_{(n)}$ are constants, while $\xi_{(k+1)}$ is independent of $\mathcal{F}_k^\xi$ and has mean zero. Hence

$$\mathbb{E}_\xi[Z_{k+1}^\xi \mid \mathcal{F}_k^\xi] = Z_k^\xi + A_{(k+1)} \mathbb{E}_\xi[\xi_{(k+1)} \mid \mathcal{F}_k^\xi] = Z_k^\xi.$$

Moreover, $|A_{(i)}| \leq 2$ makes every Z_k^ξ square-integrable. Thus $(Z_k^\xi)_{k=0}^n$ is a square-integrable martingale with respect to $(\mathcal{F}_k^\xi)_{k=0}^n$. Doob's L^2 maximal inequality gives

$$\mathbb{E}_\xi \left[\max_{0 \leq k \leq n} |Z_k^\xi|^2 \right] \leq 4 \mathbb{E}_\xi[(Z_n^\xi)^2]. \tag{121}$$

Expanding the right-hand side and using independence, $\mathbb{E}_\xi[\xi_{(i)}] = 0$, and $\mathbb{E}_\xi[\xi_{(i)}^2] = 1$ yields

$$\mathbb{E}_\xi[(Z_n^\xi)^2] = \mathbb{E}_\xi \left[\left\{ \sum_{i=1}^{n} \xi_{(i)} A_{(i)} \right\}^2 \right] = \sum_{i=1}^{n} A_{(i)}^2.$$

By Eq. (121), Jensen's inequality, and $|A_{(i)}| \leq 2$,

$$\mathbb{E}_\xi \left[\max_{0 \leq k \leq n} |Z_k^\xi| \right] \leq 2 \left(\sum_{i=1}^{n} A_{(i)}^2 \right)^{1/2} \leq 4\sqrt{n}.$$

Combining Eq. (119) and Eq. (120),

$$\mathbb{E}_\xi \left[\sup_{I \in \mathcal{I}} \left| \frac{1}{n} \sum_{i=1}^{n} \xi_i A_i \mathbf{1}\{s_i \in I\} \right| \right] \leq \frac{8}{\sqrt{n}}. \tag{122}$$

Step 3: uniform deviation for one triple (c, u, v). We derive the required high-probability bound from Eq. (122). Let $(X_i', Y_i')_{i=1}^n$ be an independent copy of the sample. Conditional Jensen's inequality yields

$$\begin{aligned}&\mathbb{E} \left[\sup_{I \in \mathcal{I}} \left| \frac{1}{n} \sum_{i=1}^{n} h_I(X_i, Y_i) - \mathbb{E}[h_I(X, Y)] \right| \right] \\ &= \mathbb{E} \left[\sup_{I \in \mathcal{I}} \left| \frac{1}{n} \sum_{i=1}^{n} h_I(X_i, Y_i) - \mathbb{E} \left[\frac{1}{n} \sum_{i=1}^{n} h_I(X_i', Y_i') \middle| (X_i, Y_i)_{i=1}^n \right] \right| \right] \\ &\leq \mathbb{E} \left[\sup_{I \in \mathcal{I}} \left| \frac{1}{n} \sum_{i=1}^{n} \{h_I(X_i, Y_i) - h_I(X_i', Y_i')\} \right| \right].\end{aligned}$$

By exchangeability of each pair $((X_i, Y_i), (X_i', Y_i'))$ and independent Rademacher signs,

$$\begin{aligned}
&\mathbb{E}\left[\sup_{I\in\mathcal{I}}\left|\frac{1}{n}\sum_{i=1}^{n}\{h_I(X_i,Y_i)-h_I(X_i',Y_i')\}\right|\right]\\
&=\mathbb{E}\left[\sup_{I\in\mathcal{I}}\left|\frac{1}{n}\sum_{i=1}^{n}\xi_i\{h_I(X_i,Y_i)-h_I(X_i',Y_i')\}\right|\right]\\
&\le\mathbb{E}\left[\sup_{I\in\mathcal{I}}\left|\frac{1}{n}\sum_{i=1}^{n}\xi_i h_I(X_i,Y_i)\right|\right]+\mathbb{E}\left[\sup_{I\in\mathcal{I}}\left|\frac{1}{n}\sum_{i=1}^{n}\xi_i h_I(X_i',Y_i')\right|\right]\\
&\le\frac{16}{\sqrt{n}},
\end{aligned}$$

where the last inequality applies Eq. (122) twice. Consequently,

$$\mathbb{E}\left[\sup_{I\in\mathcal{I}}\left|\frac{1}{n}\sum_{i=1}^{n}h_I(X_i,Y_i)-\mathbb{E}[h_I(X,Y)]\right|\right]\le\frac{16}{\sqrt{n}}. \tag{123}$$

Set

$$\Phi((X_i,Y_i)_{i=1}^n):=\sup_{I\in\mathcal{I}}\left|\frac{1}{n}\sum_{i=1}^{n}h_I(X_i,Y_i)-\mathbb{E}[h_I(X,Y)]\right|.$$

Replacing one observation changes each empirical average by at most $4/n$, because h_I takes values in $[-2,2]$. Hence it changes Φ by at most $4/n$. McDiarmid's inequality implies, for every $t>0$,

$$\mathbb{P}(\Phi-\mathbb{E}[\Phi]\ge t)\le\exp\left(-\frac{nt^2}{8}\right). \tag{124}$$

Taking $t=4\sqrt{\log(1/\delta')/(2n)}$ yields, with probability at least $1-\delta'$,

$$\sup_{I\in\mathcal{I}}|\widehat{\Gamma}_{c,u,I,v}(f)-\Gamma_{c,u,I,v}(f)|\le\frac{16}{\sqrt{n}}+4\sqrt{\frac{\log(1/\delta')}{2n}}. \tag{125}$$

Step 4: simultaneous control for all triples (c,u,v)**.** If $N_{\mathrm{cand}}=0$, both empirical and population HUC are zero and the conclusion is immediate. Otherwise set $\delta':=\delta/N_{\mathrm{cand}}$ in Eq. (125). A union bound gives, with probability at least $1-\delta$, simultaneously for all triples,

$$\sup_{I\in\mathcal{I}}|\widehat{\Gamma}_{c,u,I,v}(f)-\Gamma_{c,u,I,v}(f)|\le\frac{16}{\sqrt{n}}+4\sqrt{\frac{\log(N_{\mathrm{cand}}/\delta)}{2n}}. \tag{126}$$

Since $N_{\mathrm{cand}}\ge 1$ in this case, $\max(1,N_{\mathrm{cand}})=N_{\mathrm{cand}}$.

Finally, for any bounded real families $(x_\lambda)_\lambda$ and $(y_\lambda)_\lambda$,

$$\left|\sup_\lambda|x_\lambda|-\sup_\lambda|y_\lambda|\right|\le\sup_\lambda|x_\lambda-y_\lambda|. \tag{127}$$

Indeed, the triangle inequality gives $|x_\lambda|\le|y_\lambda|+|x_\lambda-y_\lambda|$, hence $\sup_\lambda|x_\lambda|-\sup_\lambda|y_\lambda|\le\sup_\lambda|x_\lambda-y_\lambda|$. Interchanging x and y gives the reverse direction. Applying Eq. (127) with $x_\lambda=\widehat{\Gamma}_{c,u,I,v}(f)$ and $y_\lambda=\Gamma_{c,u,I,v}(f)$, and then using Eq. (126), proves Eq. (15). □

F.2 Reduction to UC with At Most One Relevant Node

Assume $|\mathcal{V}_{\mathcal{T}}(u)|\le 1$ for every $u\in\mathcal{U}$. If $\mathcal{V}_{\mathcal{T}}(u)=\{v_u\}$, the tree decomposition contains only the contribution of v_u; if the relevant-node set is empty, the utility residual is identically zero. Therefore $\mathrm{UC}(f;\mathcal{C},\mathcal{U})=\mathrm{HUC}(f;\mathcal{C},\mathcal{U})$. Likewise, $\widehat{\mathrm{UC}}(f;\mathcal{C},\mathcal{U})=\widehat{\mathrm{HUC}}(f;\mathcal{C},\mathcal{U})$. Here $\widehat{\mathrm{UC}}$ denotes UC with the expectation replaced by its empirical average. The weight $A(X,Y)$ in the proof above then equals $c(X)\{u(P,Y)-\mu_u(P)\}$. The half-line reduction, score-ordered Rademacher partial sums, Doob's inequality, symmetrization, and concentration specialize without change to the UC proof. A flat tree recovers multiclass UC, and a two-leaf tree recovers binary UC.

F.3 Exact Empirical Maximization over All Intervals

On a fixed sample of size $n \geq 1$, fix (c, u, v) and set

$$w_i := c(X_i)\,\langle \boldsymbol{\Delta}_{u,v}(P_i), \boldsymbol{R}_v(P_i, Y_i)\rangle\,, \qquad s_i := \mu_u(P_i). \tag{128}$$

Then the empirical moment in Eq. (117) is $\widehat{\Gamma}_{c,u,I,v}(f) = \frac{1}{n}\sum_{i=1}^{n} w_i \mathbf{1}\{s_i \in I\}$. We derive an exact maximization over $I \in \mathcal{I}$ and give pseudocode that returns a maximizing interval together with its signed empirical moment.

From intervals to tied score blocks. Sort the observations by s_i. Let $m_{\mathrm{sc}} \leq n$ be the number of distinct scores, ordered as $s_{(1)} < \cdots < s_{(m_{\mathrm{sc}})}$, and define

$$G_r := \{i : s_i = s_{(r)}\}, \qquad B_r := \frac{1}{n}\sum_{i \in G_r} w_i. \tag{129}$$

All observations in G_r have the same interval membership, so their contributions must be grouped before maximization.

If an interval contains $s_{(a)}$ and $s_{(b)}$, it also contains every score between them. Thus its selected distinct scores form either the empty set or a contiguous block $s_{(a)}, \ldots, s_{(b)}$. In the latter case,

$$\widehat{\Gamma}_{c,u,I,v}(f) = \sum_{r : s_{(r)} \in I} \frac{1}{n}\sum_{i \in G_r} w_i = \sum_{r=a}^{b} B_r.$$

Conversely, every such block is selected by the closed interval $[s_{(a)}, s_{(b)}]$, including a singleton when $a = b$. The empty interval gives zero. Therefore

$$\sup_{I \in \mathcal{I}} |\widehat{\Gamma}_{c,u,I,v}(f)| = \max\left(0, \max_{1 \leq a \leq b \leq m_{\mathrm{sc}}} \left|\sum_{k=a}^{b} B_k\right|\right). \tag{130}$$

This reduction preserves every possible interval membership set on the sample, including ties.

Prefix-sum representation. Set $C_0 := 0$ and $C_b := \sum_{k=1}^{b} B_k$. For $1 \leq a \leq b \leq m_{\mathrm{sc}}$, $\sum_{k=a}^{b} B_k = \sum_{k=1}^{b} B_k - \sum_{k=1}^{a-1} B_k = C_b - C_{a-1}$. Writing $r = a-1$ gives $0 \leq r < b$. In particular, r is the position immediately before the selected block, and $\widehat{\Gamma}_{c,u,[s_{(r+1)},s_{(b)}],v}(f) = C_b - C_r$. Since $m_{\mathrm{sc}} \geq 1$ and absolute values are nonnegative, the zero in Eq. (130) may be omitted when computing the maximum value. Hence

$$\sup_{I \in \mathcal{I}} |\widehat{\Gamma}_{c,u,I,v}(f)| = \max_{1 \leq b \leq m_{\mathrm{sc}}} \max_{0 \leq r < b} |C_b - C_r|. \tag{131}$$

The outer maximum selects the right endpoint; the inner maximum selects the left endpoint for that fixed right endpoint.

Two candidates for each right endpoint. For fixed b, the value C_b does not depend on r. Using $|t| = \max(t, -t)$ gives

$$\begin{aligned}
\max_{0 \leq r < b} |C_b - C_r| &= \max_{0 \leq r < b} \max(C_b - C_r,\ C_r - C_b) \\
&= \max\left(\max_{0 \leq r < b}(C_b - C_r),\ \max_{0 \leq r < b}(C_r - C_b)\right) \\
&= \max\left(C_b - \min_{0 \leq r < b} C_r,\ \max_{0 \leq r < b} C_r - C_b\right).
\end{aligned} \tag{132}$$

The first term maximizes the signed interval sum by subtracting the smallest previous prefix sum. The second maximizes its negative by subtracting C_b from the largest previous prefix sum. Thus only these two previous prefix sums, rather than all possible left endpoints, are needed for each b.

Algorithm 2 Exact tie-preserving interval maximization for a fixed (c, u, v)

Input: $(s_i, w_i)_{i=1}^n$ from Eq. (128), with $n \geq 1$
Output: A maximizing interval, its signed moment, and its absolute value

1: Sort the pairs (s_i, w_i) by nondecreasing s_i
2: Scan the sorted pairs once to form $s_{(r)}, G_r, B_r$ in Eq. (129)
3: $C_0 \leftarrow 0$; $r_{\min} \leftarrow 0$; $r_{\max} \leftarrow 0$
4: $I_{\text{best}} \leftarrow \varnothing$; $\widehat{\Gamma}_{\text{best}} \leftarrow 0$
5: **for** $b = 1, \ldots, m_{\text{sc}}$ **do**
6: $\quad C_b \leftarrow C_{b-1} + B_b$
7: $\quad$ **if** $C_b - C_{r_{\min}} > |\widehat{\Gamma}_{\text{best}}|$ **then**
8: $\quad\quad \widehat{\Gamma}_{\text{best}} \leftarrow C_b - C_{r_{\min}}$
9: $\quad\quad I_{\text{best}} \leftarrow [s_{(r_{\min}+1)}, s_{(b)}]$
10: $\quad$ **end if**
11: $\quad$ **if** $C_{r_{\max}} - C_b > |\widehat{\Gamma}_{\text{best}}|$ **then**
12: $\quad\quad \widehat{\Gamma}_{\text{best}} \leftarrow C_b - C_{r_{\max}}$
13: $\quad\quad I_{\text{best}} \leftarrow [s_{(r_{\max}+1)}, s_{(b)}]$
14: $\quad$ **end if**
15: $\quad$ **if** $C_b < C_{r_{\min}}$ **then**
16: $\quad\quad r_{\min} \leftarrow b$
17: $\quad$ **end if**
18: $\quad$ **if** $C_b > C_{r_{\max}}$ **then**
19: $\quad\quad r_{\max} \leftarrow b$
20: $\quad$ **end if**
21: **end for**
22: **return** $I_{\text{best}}, \widehat{\Gamma}_{\text{best}}, |\widehat{\Gamma}_{\text{best}}|$

Pseudocode and correspondence with the formulas. Algorithm 2 implements Eq. (132). The indices $r_{\min}$ and $r_{\max}$ store positions attaining the smallest and largest previous prefix sums. The variables I_{best} and $\widehat{\Gamma}_{\text{best}}$ store the best interval found so far and its signed empirical moment.

Before the two candidate comparisons at right endpoint b, the stored indices satisfy

$$C_{r_{\min}} = \min_{0 \leq r < b} C_r, \qquad C_{r_{\max}} = \max_{0 \leq r < b} C_r.$$

This holds initially because the only previous prefix is $C_0 = 0$. Lines 7 and 11 compare the two terms of Eq. (132) with the best absolute value so far. In the second comparison, $C_{r_{\max}} - C_b$ is the magnitude of a negative candidate when that candidate improves the best value; line 12 therefore stores $C_b - C_{r_{\max}}$, retaining its actual sign. The corresponding interval starts at the group after the subtracted prefix, explaining the index $r_{\min} + 1$ or $r_{\max} + 1$.

Only after these comparisons do lines 15 and 18 include C_b among the possible previous prefixes. Using the old indices, the updated extreme values are $\min_{0 \leq r \leq b} C_r = \min(C_{r_{\min}}, C_b)$, $\max_{0 \leq r \leq b} C_r = \max(C_{r_{\max}}, C_b)$. The invariant therefore holds for the next right endpoint $b + 1$. Updating the extrema after the comparisons ensures that the current candidates always satisfy $r < b$.

After processing right endpoint b, the saved absolute value is

$$|\widehat{\Gamma}_{\text{best}}| = \max\left(0, \max_{1 \leq j \leq b} \max_{0 \leq r < j} |C_j - C_r|\right),$$

and its saved signed value equals $\widehat{\Gamma}_{c,u,I_{\text{best}},v}(f)$. At $b = m_{\text{sc}}$, Eq. (131) proves that the returned interval is a global empirical maximizer. Strict comparisons retain the first encountered candidate at ties; if the maximum is zero, the algorithm returns the empty interval and zero. Grouping in line 2 ensures that equal scores are never separated.

Operation count. Sorting the n pairs costs $O(n \log n)$. Line 2 groups tied scores in $O(n)$, and the prefix-extrema scan performs $O(1)$ work for each of the $m_{\text{sc}} \leq n$ distinct scores. Thus Algorithm 2 costs $O(n \log n)$ from unsorted inputs and $O(n)$ once the score order is available, without enumerating the $m_{\text{sc}}(m_{\text{sc}} + 1)/2$ nonempty blocks. Input preparation and reuse of the score order are treated in Appendix F.4.

F.4 Computational Comparison of Exact UC and HUC Auditing

We compare arithmetic operation counts for exact empirical auditing on the same hierarchical dataset. Model inference is excluded, and evaluating one value $u(P_i, y)$ or one subgroup weight $c(X_i)$ is treated as constant time. Let n be the audit-sample size, $K = |\mathcal{Y}|$ the number of leaves, d_v the number of children of internal node v, and

$$|E_{\mathcal{T}}| = \sum_{v \in V^\circ} d_v = |V| - 1 \tag{133}$$

be the number of tree edges. For $u \in \mathcal{U}$, define the maximum number of relevant nodes on a root-to-leaf path by

$$D_u := \max_{y \in \mathcal{Y}} |\operatorname{Path}(y) \cap \mathcal{V}_{\mathcal{T}}(u)|, \qquad D := \max_{u \in \mathcal{U}} D_u. \tag{134}$$

We divide the computation into three stages: **1) precomputation of the required statistics, 2) sorting the predicted utility scores, and 3) maximization over all intervals.** We derive the cost of each stage for UC and HUC below. All interval maxima are computed by the exact tie-preserving scan of Appendix F.3.

Computational complexity of UC

1) Precomputation of the required statistics. For each observation i and utility u, UC uses the predicted utility score and realized utility residual

$$s_{i,u} := \mu_u(P_i) = \sum_{y \in \mathcal{Y}} P_{i,y} u(P_i, y), \qquad a_{i,u} := u(P_i, Y_i) - s_{i,u}. \tag{135}$$

For a general dense utility, computing the score requires evaluating the K leaf utility values and their weighted sum, at a cost of $O(K)$ per (i, u). The residual is then obtained in constant time from these values. Thus precomputation over all observations and utilities costs $O(n|\mathcal{U}|K)$.

2) Sorting the predicted utility scores. For each utility u, sort the n scores $s_{1,u}, \ldots, s_{n,u}$ once. This costs $O(n \log n)$ per utility and $O(n|\mathcal{U}| \log n)$ overall. Because $s_{i,u} = \mu_u(P_i)$ does not depend on c, the order is shared across all $c \in \mathcal{C}$.

3) Maximization over all intervals. For fixed (c, u), run Algorithm 2 on $w_i = c(X_i)a_{i,u}$ in the shared score order. The tie grouping and scan cost $O(n)$ per (c, u), hence $O(n|\mathcal{C}||\mathcal{U}|)$ over all subgroup weights and utilities.

Combining the three stages gives

$$T_{\mathrm{UC}} = O(n|\mathcal{U}|K + n|\mathcal{U}| \log n + n|\mathcal{C}||\mathcal{U}|)\,. \tag{136}$$

If the scores and realized utility residuals in Eq. (135) are already available, the cost of Step 1 is omitted.

Computational complexity of HUC

1) Precomputation of the required statistics. HUC uses the same leaf-level utility values as UC, together with the predicted mass and mean utility of each subtree. For each (i, u), initialize the leaf values by $\pi_y(P_i) = P_{i,y}$, $N_{u,y}(P_i) = P_{i,y}u(P_i, y)$. Then traverse the tree from the leaves to the root, using the recursions in Lemma C.1:

$$\pi_v(P_i) = \sum_{j=1}^{d_v} \pi_{v_j}(P_i), \qquad N_{u,v}(P_i) = \sum_{j=1}^{d_v} N_{u,v_j}(P_i).$$

These values give the subtree means, branch probabilities, and local utility contrasts: $\mu_{u,v}(P_i) = \frac{N_{u,v}(P_i)}{\pi_v(P_i)}$, $q_{v,j}(P_i) = \frac{\pi_{v_j}(P_i)}{\pi_v(P_i)}$, $\Delta_{u,v,j}(P_i) = \mu_{u,v_j}(P_i) - \mu_{u,v}(P_i)$. At each internal node, we reuse the already computed child aggregates rather than summing its descendant-leaf values from scratch. The work at node v is $O(d_v)$, so the total tree computation costs $O\big(\sum_{v \in V^\circ} d_v\big) = O(|E_{\mathcal{T}}|)$. Including the leaf utility evaluation, precomputation costs $O(K + |E_{\mathcal{T}}|)$ per (i, u) and $O(n|\mathcal{U}|(K + |E_{\mathcal{T}}|))$ over all observations and utilities.

2) Sorting the predicted utility scores. HUC uses the same score $s_{i,u} = \mu_u(P_i)$ as UC, so one sort per utility costs $O(n|\mathcal{U}| \log n)$. The order depends on neither c nor v and is therefore shared across all subgroup weights and relevant nodes.

3) Maximization over all intervals. For fixed (c, u), make one pass in this shared order while maintaining the Algorithm 2 state for the relevant nodes. By Lemma C.2, observation i can contribute only at nodes in $\mathrm{Path}(Y_i) \cap \mathcal{V}_\mathcal{T}(u)$, of which there are at most D_u; each such contribution is available in constant time from the subtree means computed in Step 1.

Equal-score observations are accumulated as one block, and the node-wise prefix states are advanced only after the block is complete, so tied scores are never split. Nodes not touched by the block retain the same state, while the global maximum is updated from the touched nodes. The scan therefore processes only pairs consisting of an observation and a relevant node on its realized path. Their total number is

$$\sum_{i=1}^{n} \left|\mathrm{Path}(Y_i) \cap \mathcal{V}_\mathcal{T}(u)\right| \le nD_u.$$

Adding each contribution and performing each node-wise update at the end of a group take constant time, and the number of the latter updates is no larger than the number of these pairs. Exact interval maximization thus costs $O(nD_u)$ per (c, u) and $O\big(n|\mathcal{C}| \sum_{u\in\mathcal{U}} D_u\big)$ over all subgroup weights and utilities. Nodes not reached by any observation have empirical moment zero and need not be scanned.

Combining the three stages gives

$$T_{\mathrm{HUC}} = O\left(n|\mathcal{U}|(K + |E_\mathcal{T}|) + n|\mathcal{U}| \log n + n|\mathcal{C}| \sum_{u\in\mathcal{U}} D_u\right). \tag{137}$$

Contracting unary nodes. A node with only one child has identically zero local contribution, so contracting such nodes leaves the audit value unchanged. After contraction every internal node has at least two children. Writing $M := |V^\circ|$, we have $2M \le \sum_{v\in V^\circ} d_v = |E_\mathcal{T}| = K + M - 1$. Therefore $M \le K - 1$, $|E_\mathcal{T}| \le 2K - 2$, and the tree precomputation in Step 1 is also linear in K. Thus Eq. (137) becomes

$$\begin{aligned} T_{\mathrm{HUC}} &= O\left(n|\mathcal{U}|K + n|\mathcal{U}| \log n + n|\mathcal{C}| \sum_{u\in\mathcal{U}} D_u\right) \\ &= O(n|\mathcal{U}|K + n|\mathcal{U}| \log n + n|\mathcal{C}||\mathcal{U}|D)\,, \end{aligned} \tag{138}$$

where the last bound uses $\sum_{u\in\mathcal{U}} D_u \le |\mathcal{U}|D$.

Comparison. Table 3 summarizes the three stages. UC does not traverse internal nodes: its $O(K)$ term comes from computing the predicted mean of a dense utility over the leaves. HUC uses the same leaf-level quantities and adds a tree traversal that is also linear in K on a reduced tree, together with path-sparse interval accumulation. Consequently, for a fixed finite subgroup class, both exact audits have leading order

$$O(n|\mathcal{U}|K + n|\mathcal{U}| \log n)\,. \tag{139}$$

They are not identical in every computational regime. If the predicted utility scores and all local subtree quantities have already been computed, the remaining scan costs are

$$T_{\mathrm{UC}}^{\mathrm{scan}} = O(n|\mathcal{U}| \log n + n|\mathcal{C}||\mathcal{U}|)\,, \qquad T_{\mathrm{HUC}}^{\mathrm{scan}} = O\left(n|\mathcal{U}| \log n + n|\mathcal{C}| \sum_{u\in\mathcal{U}} D_u\right). \tag{140}$$

Thus, the additional cost of localization is controlled by the number of relevant nodes on the realized root-to-leaf paths, rather than by enumerating all intervals separately for every node.

Sparse utilities and working storage. Let $s_u := |\mathcal{V}_\mathcal{T}(u)|$. Eq. (137) is an end-to-end bound for a general dense utility, for which the leaf values must first be aggregated through the tree. If a utility is supplied directly by a cut-level representation or its required subtree means have already been

Table 3: Stage-wise operation counts for exact empirical UC and HUC auditing on the same sample, for general dense utilities. Model inference is excluded; each total is the sum of the three stages.

Computational stage	UC	HUC
1) Precomputation of the required statistics	$O(n\|\mathcal{U}\|K)$	$O(n\|\mathcal{U}\|(K+\|E_{\mathcal{T}}\|))$
2) Sorting the predicted utility scores	$O(n\|\mathcal{U}\|\log n)$	$O(n\|\mathcal{U}\|\log n)$
3) Maximization over all intervals	$O(n\|\mathcal{C}\|\|\mathcal{U}\|)$	$O\big(n\|\mathcal{C}\|\sum_{u\in\mathcal{U}} D_u\big)$

cached, the $K + |E_{\mathcal{T}}|$ preprocessing term can be replaced by the size of that active representation. Processing one utility at a time requires $O(n + |\mathcal{C}|s_u)$ working scalar storage after the score order has been formed; retaining all utilities simultaneously gives $O(n|\mathcal{U}| + |\mathcal{C}|\sum_u s_u)$. These storage choices do not change the operation counts above.

G Post-processing and HUC

The post-processing guarantee for UC bounds the risk improvement achievable by a nondecreasing transformation of predicted mean utility (Hegazy et al., 2026; Rossellini et al., 2025). We apply the same comparison to hierarchically aggregated utilities derived from the original target utility. Specifically, we stop refinement at selected positions in the label tree and average the utility over the leaves below each stopping position. Small HUC for the original utility then controls, for each such aggregation, the excess risk over the best monotone post-processing of the common predicted-utility score.

Branch selection and hierarchical utility aggregation. Let $\mathcal{S} \subseteq V^\circ$ be the set of internal nodes at which branching continues. Whenever a node belongs to $\mathcal{S}$, all its strict internal ancestors must also belong to $\mathcal{S}$. Thus choosing $\mathcal{S}$ determines where the tree is truncated for each leaf. Follow the path from the root to a leaf y: continue to the child on that path while the current node belongs to $\mathcal{S}$, and stop at the first node outside $\mathcal{S}$. Denote this stopping node by $b_{\mathcal{S}}(y)$.

If the path contains any node in $\mathcal{S}$, its boundary $b_{\mathcal{S}}(y)$ is the child toward y of the last such node on that path. The boundary itself is not in $\mathcal{S}$. When $\mathcal{S} = \varnothing$, the procedure stops at the root; when $\mathcal{S} = V^\circ$, it reaches every original leaf. Different paths may stop at different depths.

Below the boundary, individual leaves are no longer distinguished, and the original utility is averaged using the predictive distribution within that subtree. Define the resulting *hierarchically aggregated utility* by

$$u_{\mathcal{S}}(p, y) := \mu_{u,b_{\mathcal{S}}(y)}(p) = \frac{\sum_{z\in\mathcal{Y}_{b_{\mathcal{S}}(y)}} p_z u(p, z)}{\pi_{b_{\mathcal{S}}(y)}(p)}. \tag{141}$$

Leaves with the same boundary receive the same utility value. This is not an arbitrarily specified new payoff: it is determined by the original utility u, the prediction p, and the branch set $\mathcal{S}$. In particular, $u_{\varnothing}(p, y) = \mu_u(p)$, $u_{V^\circ}(p, y) = u(p, y)$. Using no branches gives the overall predicted mean, whereas retaining every branch recovers the original leaf utility.

Lemma G.1 (Decomposition and predicted mean of the aggregated utility)**.** *For every positive prediction vector p and leaf y, we have $u_{\mathcal{S}}(p, y) \in [-1, 1]$ and*

$$u_{\mathcal{S}}(p, y) - \mu_u(p) = \sum_{v\in\mathcal{S}} \langle \boldsymbol{\Delta}_{u,v}(p), \boldsymbol{R}_v(p, y)\rangle\,, \tag{142}$$

$$\mu_{u_{\mathcal{S}}}(p) = \mu_u(p). \tag{143}$$

Proof. If the path to y proceeds from v to v_j, Lemma C.2 gives $\langle \boldsymbol{\Delta}_{u,v}(p), \boldsymbol{R}_v(p, y)\rangle = \mu_{u,v_j}(p) - \mu_{u,v}(p)$. By the definition of the boundary, the nodes where branching occurs before reaching $b_{\mathcal{S}}(y)$ are exactly the nodes of $\mathcal{S}$ on the path to y. Summing their contributions cancels every intermediate subtree mean, leaving the difference between the boundary and root means. Off-path contributions are zero, so

$$\sum_{v\in\mathcal{S}} \langle \boldsymbol{\Delta}_{u,v}(p), \boldsymbol{R}_v(p, y)\rangle = \mu_{u,b_{\mathcal{S}}(y)}(p) - \mu_{u,\rho}(p) = u_{\mathcal{S}}(p, y) - \mu_u(p).$$

The subtree mean is a weighted average of utility values in $[-1, 1]$, which also proves boundedness.

Under the predictive distribution $\widetilde{Y} \sim p$, each residual component satisfies $\mathbb{E}_{\widetilde{Y}\sim p}[R_{v,j}(p, \widetilde{Y})] = \pi_{v_j}(p) - q_{v,j}(p)\pi_v(p) = 0$. Taking predictive expectation in Eq. (142) therefore gives

$$\mu_{u_{\mathcal{S}}}(p) - \mu_u(p) = \sum_{v\in\mathcal{S}} \left\langle \boldsymbol{\Delta}_{u,v}(p), \mathbb{E}_{\widetilde{Y}\sim p}[\boldsymbol{R}_v(p, \widetilde{Y})] \right\rangle = 0. \qquad \square$$

The equality of means is under the predictive distribution p; the original and aggregated utilities need not have equal means under the actual label distribution.

Monotone post-processing of the common score. For a threshold $t \in [-1, 1]$ and a binary rule $\varphi : \mathcal{X} \to \{0, 1\}$, define

$$\mathcal{R}_{u,\mathcal{S}}(\varphi; t) := \mathbb{E}[|u_{\mathcal{S}}(P, Y) - t|\, \mathbf{1}\{\varphi(X) \neq \mathbf{1}\{u_{\mathcal{S}}(P, Y) \geq t\}\}]\,. \tag{144}$$

A wrong decision about whether the aggregated utility is at least the threshold incurs its distance from that threshold. The expectation is under the actual joint distribution of (X, Y).

Lemma G.1 shows that the predicted mean of the aggregated utility is still $\mu_u(P)$. Accordingly, define the baseline rule and its post-processed counterpart by

$$\varphi^0_{u,t}(X) := \mathbf{1}\{\mu_u(P) \geq t\}, \qquad \varphi^h_{u,t}(X) := \mathbf{1}\{h(\mu_u(P)) \geq t\}, \tag{145}$$

where $h : [-1, 1] \to [-1, 1]$ is nondecreasing. The post-processing gap is

$$\mathrm{Gap}^{\mathrm{mono}}_{t,\mathcal{S}}(f, u) := \mathcal{R}_{u,\mathcal{S}}(\varphi^0_{u,t}; t) - \inf_{h \text{ nondecreasing}} \mathcal{R}_{u,\mathcal{S}}(\varphi^h_{u,t}; t). \tag{146}$$

Here $f, u, \mathcal{S}, t$ remain fixed, and only h is varied.

The post-processing operation and the class of decision rules are the same as for UC applied to the original utility. For the same t, h, the binary decisions do not depend on $\mathcal{S}$. What changes is whether those decisions are evaluated using the original leaf utility $u(P, Y)$ or the aggregated utility $u_{\mathcal{S}}(P, Y)$. Tree truncation specifies the granularity of the evaluation utility, not a replacement of the prediction rule by node-specific threshold decisions.

Post-processing guarantee for aggregated utilities.

Theorem G.2 (Monotone post-processing guarantee for hierarchical utility aggregation)**.** *For every target utility u, threshold $t \in [-1, 1]$, and set $\mathcal{S} \subseteq V^\circ$ satisfying the ancestor condition above,*

$$\mathrm{Gap}^{\mathrm{mono}}_{t,\mathcal{S}}(f, u) \leq 2\ \mathrm{UC}(f; \{1\}, \{u_{\mathcal{S}}\}) \leq 2|\mathcal{S} \cap \mathcal{V}_{\mathcal{T}}(u)|\ \mathrm{HUC}(f; \{1\}, \{u\}). \tag{147}$$

The first inequality is the UC post-processing guarantee applied to $u_{\mathcal{S}}$. The second bounds its UC error using the node-wise errors of the original utility u. We make these two steps explicit below.

Proof. Let $Z := \mu_u(P)$ and $U := u_{\mathcal{S}}(P, Y)$. Lemma G.1 gives $Z = \mu_{u_{\mathcal{S}}}(P)$, $U, Z \in [-1, 1]$, and $U - Z = \sum_{v\in\mathcal{S}} \langle \boldsymbol{\Delta}_{u,v}(P), \boldsymbol{R}_v(P, Y)\rangle$. For every $I \in \mathcal{I}$, multiplication by $\mathbf{1}\{Z \in I\}$ and expectation yield

$$\mathbb{E}[\mathbf{1}\{Z \in I\}(U - Z)] = \sum_{v\in\mathcal{S}\cap\mathcal{V}_{\mathcal{T}}(u)} \Gamma_{1,u,I,v}(f). \tag{148}$$

We omit nodes whose local utility contrast is identically zero. The triangle inequality and the definition of HUC imply

$$\mathrm{UC}(f; \{1\}, \{u_{\mathcal{S}}\}) = \sup_{I\in\mathcal{I}} \left| \sum_{v\in\mathcal{S}\cap\mathcal{V}_{\mathcal{T}}(u)} \Gamma_{1,u,I,v}(f) \right| \leq |\mathcal{S} \cap \mathcal{V}_{\mathcal{T}}(u)|\ \mathrm{HUC}(f; \{1\}, \{u\}), \tag{149}$$

which proves the second inequality.

For the first inequality, every $a \in \{0, 1\}$ satisfies the pointwise identity

$$|U - t|\mathbf{1}\{a \neq \mathbf{1}\{U \geq t\}\} = \frac{1}{2}|U - t| + \frac{1}{2}(1 - 2a)(U - t). \tag{150}$$

For $a = 0$, the right-hand side is $U - t$ when $U \geq t$ and zero otherwise; for $a = 1$, it is $t - U$ when $U < t$ and zero otherwise. Thus

$$\mathcal{R}_{u,\mathcal{S}}(\varphi; t) = \frac{1}{2}\mathbb{E}|U - t| + \frac{1}{2}\mathbb{E}[(1 - 2\varphi(X))(U - t)].$$

The first term does not depend on the rule and cancels in the risk difference, giving

$$\mathcal{R}_{u,\mathcal{S}}(\varphi^0_{u,t}; t) - \mathcal{R}_{u,\mathcal{S}}(\varphi^h_{u,t}; t) = \mathbb{E}[(U - t)\{\varphi^h_{u,t}(X) - \varphi^0_{u,t}(X)\}]. \tag{151}$$

Let $E_+ := \{Z < t,\ h(Z) \geq t\}$ and $E_- := \{Z \geq t,\ h(Z) < t\}$ be the events on which the decision changes from 0 to 1 and from 1 to 0, respectively. Since $\varphi^h_{u,t}(X) - \varphi^0_{u,t}(X) = \mathbf{1}(E_+) - \mathbf{1}(E_-)$,

$$\mathcal{R}_{u,\mathcal{S}}(\varphi^0_{u,t}; t) - \mathcal{R}_{u,\mathcal{S}}(\varphi^h_{u,t}; t) = \mathbb{E}[(U - t)\mathbf{1}(E_+)] + \mathbb{E}[(t - U)\mathbf{1}(E_-)].$$

On E_+, $Z < t$ gives $U - t = (U - Z) + (Z - t) \leq U - Z$. On E_-, $Z \geq t$ gives $t - U = (Z - U) + (t - Z) \leq -(U - Z)$. Therefore,

$$\mathcal{R}_{u,\mathcal{S}}(\varphi^0_{u,t}; t) - \mathcal{R}_{u,\mathcal{S}}(\varphi^h_{u,t}; t) \leq |\mathbb{E}[(U - Z)\mathbf{1}(E_+)]| + |\mathbb{E}[(U - Z)\mathbf{1}(E_-)]|.$$

Because h is nondecreasing, each of the score sets $\{z \in [-1, 1] : z < t,\ h(z) \geq t\}$ and $\{z \in [-1, 1] : z \geq t,\ h(z) < t\}$ is an interval, possibly empty. Each term is consequently at most $\sup_{I \in \mathcal{I}} |\mathbb{E}[\mathbf{1}\{Z \in I\}(U - Z)]| = \mathrm{UC}(f; \{1\}, \{u_{\mathcal{S}}\})$. Taking the supremum of the improvement over all nondecreasing h proves the first inequality. Combining it with Eq. (149) proves Eq. (147). □

Subtree-mean calibration and refinement. The same decomposition also yields the subgroup-weighted calibration guarantee used in the remark following Definition 3.2.

Proposition G.3 (Calibration under fixed tree boundaries)**.** *For every subgroup class $\mathcal{C}$, target utility u, and set $\mathcal{S} \subseteq V^\circ$ satisfying the ancestor condition above,*

$$\mathrm{UC}(f; \mathcal{C}, \{u_{\mathcal{S}}\}) \leq |\mathcal{S} \cap \mathcal{V}_{\mathcal{T}}(u)|\ \mathrm{HUC}(f; \mathcal{C}, \{u\}). \tag{152}$$

These bounds hold simultaneously for all such fixed tree boundaries. Moreover, if $v \notin \mathcal{S}$ is an internal boundary node, meaning that all its strict internal ancestors are in $\mathcal{S}$, then $\mathcal{S}^+ := \mathcal{S} \cup \{v\}$ also satisfies this condition and

$$\begin{aligned} &\{u_{\mathcal{S}^+}(p, y) - \mu_{u_{\mathcal{S}^+}}(p)\} - \{u_{\mathcal{S}}(p, y) - \mu_{u_{\mathcal{S}}}(p)\} \\ &\quad = u_{\mathcal{S}^+}(p, y) - u_{\mathcal{S}}(p, y) = \langle \boldsymbol{\Delta}_{u,v}(p), \boldsymbol{R}_v(p, y) \rangle. \end{aligned} \tag{153}$$

Its moment under the same audit conditions is

$$\mathbb{E}[c(X)\mathbf{1}\{\mu_u(P) \in I\}\{u_{\mathcal{S}^+}(P, Y) - u_{\mathcal{S}}(P, Y)\}] = \Gamma_{c,u,I,v}(f). \tag{154}$$

Proof. Lemma G.1 gives $u_{\mathcal{S}}(p, y) \in [-1, 1]$ and $\mu_{u_{\mathcal{S}}}(p) = \mu_u(p)$. Since the tree boundary is fixed, $u_{\mathcal{S}}$ is a measurable utility of the same permitted form as u. Its predicted-utility intervals therefore select exactly the same observations. By Eq. (142), for every $c \in \mathcal{C}$ and $I \in \mathcal{I}$,

$$\mathbb{E}[c(X)\mathbf{1}\{\mu_{u_{\mathcal{S}}}(P) \in I\}\{u_{\mathcal{S}}(P, Y) - \mu_{u_{\mathcal{S}}}(P)\}] = \sum_{v \in \mathcal{S} \cap \mathcal{V}_{\mathcal{T}}(u)} \Gamma_{c,u,I,v}(f).$$

The finite sum may be interchanged with expectation, and nodes outside $\mathcal{V}_{\mathcal{T}}(u)$ contribute zero. Thus its absolute value is at most $\sum_{v \in \mathcal{S} \cap \mathcal{V}_{\mathcal{T}}(u)} |\Gamma_{c,u,I,v}(f)| \leq |\mathcal{S} \cap \mathcal{V}_{\mathcal{T}}(u)|\ \mathrm{HUC}(f; \mathcal{C}, \{u\})$. Taking the supremum over c, I proves Eq. (152). The argument applies to every $\mathcal{S}$ satisfying the ancestor condition with the same right-hand HUC error, proving simultaneous control.

For the refinement, the assumption on v ensures that only its boundary subtree is expanded, replacing its value $\mu_{u,v}(p)$ by $\mu_{u,v_j}(p)$ on each child subtree. Subtracting Eq. (142) for $\mathcal{S}$ from the same identity for $\mathcal{S}^+$ leaves exactly the term indexed by v. Both predicted means equal $\mu_u(p)$ by Lemma G.1, proving Eq. (153). Multiplication by $c(X)\mathbf{1}\{\mu_u(P) \in I\}$ and expectation give Eq. (154). □

Thus a node contribution is precisely the additional calibration residual from one refinement, and its audit moment evaluates the corresponding change in mean utility under the common audit conditions.

Relation to UC. For the original utility, the tree decomposition gives

$$\mathrm{UC}(f;\{1\},\{u\}) = \sup_{I\in\mathcal{I}} \left| \sum_{v\in\mathcal{V}_{\mathcal{T}}(u)} \Gamma_{1,u,I,v}(f) \right|.$$

For $u_{\mathcal{S}}$, the sum is restricted to nodes in $\mathcal{S}$, as in Eq. (149). Both use intervals of the same score $\mu_u(P)$. Positive and negative contributions may cancel in the full sum without making every selected partial sum small. Small UC for the original utility alone therefore need not imply small post-processing gaps for its aggregated utilities. HUC controls the individual contributions and hence provides the guarantee for every fixed aggregation, with the relevant-node factor in Theorem G.2.

This does not mean that UC cannot evaluate aggregated utilities: each $u_{\mathcal{S}}$ can be included explicitly in its utility class. The distinction is that the right-hand side of our guarantee uses HUC only for the original utility u, while controlling the post-processing gaps for the family of aggregated utilities derived from it. With $\mathcal{S} = V^{\circ}$, the utility and the gap reduce to those for the original leaf utility.

Interpretation as excess risk. If $\mathrm{HUC}(f;\{1\},\{u\}) \le \varepsilon$, Theorem G.2 can be rearranged as

$$\mathcal{R}_{u,\mathcal{S}}(\varphi^0_{u,t};t) \le \inf_{h \text{ nondecreasing}} \mathcal{R}_{u,\mathcal{S}}(\varphi^h_{u,t};t) + 2|\mathcal{S}\cap\mathcal{V}_{\mathcal{T}}(u)|\,\varepsilon. \tag{155}$$

Thus, for each aggregated utility, the current threshold rule is close to the best rule obtainable by monotone post-processing of the same score. Its risk is small when both this benchmark risk and the error term are small. Small HUC alone does not guarantee small absolute risk, nor does it imply optimality among arbitrary decision rules.

H Empirical HUC-Boost and Fixed-Step Sample Complexity

HUC-Boost is implemented on a finite calibration sample using empirical audit moments. The implementation supports adaptive and fixed node-logit step sizes. The adaptive step size uses the empirical analogue $\widehat{\Gamma}_t/\widehat{\Lambda}_t$ of the population step in Algorithm 1; this is the HUC-Boost step used for the main experimental method. The fixed step size uses a prescribed signed magnitude and gives the HUC-Boost (0.05) comparison when that magnitude is 0.05. Algorithm 3 records both choices. The sample-complexity theorems in this appendix are stated for the fixed-step specialization. We then give the corresponding two-stage construction with an initializer learned from the same sample and finally describe the initialization models used in the experiments.

H.1 Fixed-Step Sample Complexity of HUC-Boost

Let the audit sample be $S = (Z_1,\dots,Z_n) \sim \mathbb{P}^n$, where $Z_i = (X_i, Y_i)$. The tree, the finite classes $\mathcal{C}$,$\mathcal{U}$, and the functions they contain are fixed independently of S. We use the empirical audit moment in Eq. (117), the empirical HUC from Section 3.1, and the empirical log loss defined in Eq. (104). Maximization over intervals in empirical HUC is computed by the finite-point procedure in Appendix F.3, and ties are resolved by the fixed order in Appendix A.1.

The number of audit candidates is N_{cand} from Eq. (42). If $N_{\mathrm{cand}} = 0$, HUC is identically zero, so we assume $N_{\mathrm{cand}} \ge 1$ below. The following algorithm uses the same one-node logit direction as the population Algorithm 1, but replaces the population moment and curvature by their empirical counterparts.

When the adaptive step size is used, $|\widehat{\Gamma}_t| > \tau$ implies $\widehat{\Lambda}_t > 0$ by the empirical analogue of Lemma E.4. The experiments use this step size for HUC-Boost and for the HUC-Boost stage after a parametric recalibrator; HUC-Boost (0.05) uses the fixed step size. Their candidate traversal, stopping tolerance, and update budgets are specified in the experimental appendices.

For the theoretical results below, specialize Algorithm 3 to the fixed step size with $\tau = 3\varepsilon/4$ and

$$\alpha_{\mathrm{fix}} = \alpha_\varepsilon := \frac{\varepsilon}{2}, \qquad 0 < \varepsilon \le 1. \tag{156}$$

Assume that the positive initial predictor f_0 is fixed independently of S, and that a deterministic constant $D_{\mathrm{fix}} \ge 0$ satisfies

$$\widehat{\mathcal{L}}_{\log,S}(f_0) \le D_{\mathrm{fix}} \quad \text{almost surely.} \tag{157}$$

Algorithm 3 Empirical HUC-Boost with adaptive and fixed step sizes

Input: Calibration sample S, positive initial predictor f_0, fixed tree and classes $\mathcal{C}, \mathcal{U}, \mathcal{I}$, audit threshold $\tau > 0$, choice of adaptive or fixed step size, and $\alpha_{\mathrm{fix}} > 0$ for the fixed step size

$z_{0,v,j}(x) \leftarrow \log q_{v,j}(f_0(x))$ for all v, j; $t \leftarrow 0$

while there exists a candidate satisfying $|\widehat{\Gamma}_{c,u,I,v}(f_t)| > \tau$ **do**

  Select any such $a_t = (c_t, u_t, I_t, v_t)$

  $\widehat{\Gamma}_t \leftarrow \widehat{\Gamma}_{c_t,u_t,I_t,v_t}(f_t)$; $\sigma_t \leftarrow \mathrm{sgn}(\widehat{\Gamma}_t)$

  $\boldsymbol{h}_t(x) \leftarrow c_t(x)\mathbf{1}\{\mu_{u_t}(f_t(x)) \in I_t\}\boldsymbol{\Delta}_{u_t,v_t}(f_t(x))$

  if the adaptive step size is used **then**

    $\widehat{\Lambda}_t \leftarrow \dfrac{1}{4n}\displaystyle\sum_{i=1}^{n} J_{v_t}(Y_i)\{\max_j h_{t,j}(X_i) - \min_j h_{t,j}(X_i)\}^2, \qquad \alpha_t \leftarrow \widehat{\Gamma}_t/\widehat{\Lambda}_t$

  else

    $\alpha_t \leftarrow \sigma_t \alpha_{\mathrm{fix}}$

  end if

  $\boldsymbol{z}_{t+1,v_t}(x) \leftarrow \boldsymbol{z}_{t,v_t}(x) + \alpha_t \boldsymbol{h}_t(x)$ for all x; keep all other nodes fixed

  Reconstruct f_{t+1} from the softmax branch probabilities and path products; $t \leftarrow t+1$

end while

return $\widehat{f} = f_t$

For example, this condition holds if $f_0(x)_y \geq e^{-D_{\mathrm{fix}}}$ for every (x, y).

Theorem H.1 (Termination of fixed-step HUC-Boost from a fixed initializer)**.** *For every fixed sample, Algorithm 3 in the fixed-step specialization above terminates after finitely many updates. If* N_{upd} *denotes the number of updates, then*

$$N_{\mathrm{upd}} \leq \left\lceil \frac{4\widehat{\mathcal{L}}_{\log,S}(f_0)}{\varepsilon^2} \right\rceil, \qquad \widehat{\mathrm{HUC}}(\widehat{f}; \mathcal{C}, \mathcal{U}) \leq \frac{3\varepsilon}{4}. \tag{158}$$

Under Eq. (157)*, the deterministic update bound*

$$T_\varepsilon^{\mathrm{fix}} := \left\lceil \frac{4D_{\mathrm{fix}}}{\varepsilon^2} \right\rceil \tag{159}$$

may be used.

Theorem H.2 (Population HUC guarantee for fixed-step HUC-Boost from a fixed initializer)**.** *Run Algorithm 3 in the fixed-step specialization above, assume Eq.* (157)*, and let* $0 < \delta < 1$*. If*

$$n \geq \frac{512}{\varepsilon^2}\left[(T_\varepsilon^{\mathrm{fix}} + 1)\log(2N_{\mathrm{cand}}(2n+1)^2) + \log\frac{1}{\delta}\right], \tag{160}$$

then, with probability at least $1 - \delta$ *over the audit sample,*

$$\mathrm{HUC}(\widehat{f}; \mathcal{C}, \mathcal{U}) \leq \varepsilon. \tag{161}$$

By Eq. (159) and $\lceil x \rceil \leq x + 1$, $T_\varepsilon^{\mathrm{fix}} + 1 \leq \frac{4D_{\mathrm{fix}}}{\varepsilon^2} + 2$. Hence Eq. (160) is implied by

$$n \geq \frac{512}{\varepsilon^2}\left[\left(\frac{4D_{\mathrm{fix}}}{\varepsilon^2} + 2\right)\log(2N_{\mathrm{cand}}(2n+1)^2) + \log\frac{1}{\delta}\right].$$

When D_{fix} and N_{cand} do not depend on ε, a sufficient condition with no n on the right-hand side is given in Appendix H.5, especially Corollary H.13. A sufficient sample-size order is

$$n = \widetilde{O}\left(\frac{D_{\mathrm{fix}} + 1}{\varepsilon^4} + \frac{\log(1/\delta)}{\varepsilon^2}\right).$$

In particular, for fixed δ, $n = \widetilde{O}(\varepsilon^{-4})$ is sufficient.

H.2 Fixed-Step HUC-Boost with a Learned Initializer

We now consider a two-stage procedure that learns a finite-dimensional softmax recalibrator from the same sample and then runs Algorithm 3 from its output. The experimental two-stage methods use the adaptive step size. Theorems H.4 and H.5 below analyze the fixed step size with $\tau = 3\varepsilon/4$ and $\alpha_{\mathrm{fix}} = \alpha_\varepsilon$.

Assumption H.3 (Sample-dependent initial recalibration model)**.** *Let $f_{0,\theta}(x) := \mathrm{softmax}(s_\theta(x))$, where $\theta \in \mathbb{R}^d$ and $s_\theta(x) \in \mathbb{R}^K$ is finite. Fix constants $R_\theta, L_\theta > 0$ and $D_0 \geq 0$. A measurable initial fitting rule A returns $\widehat{\theta} = \mathsf{A}(S)$ and satisfies, almost surely,*

$$\|\widehat{\theta}\|_\infty \leq R_\theta, \qquad \widehat{\mathcal{L}}_{\log,S}(f_{0,\widehat{\theta}}) \leq D_0. \tag{162}$$

Assume further that, for every $\theta, \theta' \in \mathbb{R}^d$,

$$\sup_x \|s_\theta(x) - s_{\theta'}(x)\|_\infty \leq L_\theta \|\theta - \theta'\|_\infty. \tag{163}$$

Set

$$a_\varepsilon := \frac{\varepsilon^2}{2L_\theta}, \qquad \Lambda_\varepsilon := d \log\left(1 + \frac{R_\theta}{a_\varepsilon}\right) = d \log\left(1 + \frac{2R_\theta L_\theta}{\varepsilon^2}\right). \tag{164}$$

Draw $\Xi \sim \mathrm{Unif}([-a_\varepsilon, a_\varepsilon]^d)$ independently of S, and let $\Theta_0 := \widehat{\theta} + \Xi$.

Algorithm 4 Two-stage empirical HUC-Boost with an initializer learned from the same sample

Input: S, initial fitting rule A, constants in Assumption H.3, $0 < \varepsilon \leq 1$, audit threshold τ, choice of adaptive or fixed step size, and α_{fix} for the fixed step size

1: $\widehat{\theta} \leftarrow \mathsf{A}(S)$
2: Draw $\Xi \sim \mathrm{Unif}([-a_\varepsilon, a_\varepsilon]^d)$ independently of S
3: $\Theta_0 \leftarrow \widehat{\theta} + \Xi$; $f_0 \leftarrow f_{0,\Theta_0}$
4: Run Algorithm 3 from f_0 using the same sample S, threshold τ, and the selected step-size rule
5: **return** its output $\widehat{f}$ and Θ_0

Theorem H.4 (Termination of fixed-step two-stage HUC-Boost from a learned initializer)**.** *Under Assumption H.3, Algorithm 4 using the fixed step size with $\tau = 3\varepsilon/4$ and $\alpha_{\mathrm{fix}} = \alpha_\varepsilon$ terminates almost surely and satisfies*

$$N_{\mathrm{upd}} \leq T_\varepsilon^{\mathrm{learn}} := \left\lceil \frac{4D_0}{\varepsilon^2} \right\rceil + 4, \qquad \widehat{\mathrm{HUC}}(\widehat{f}; \mathcal{C}, \mathcal{U}) \leq \frac{3\varepsilon}{4}. \tag{165}$$

Theorem H.5 (Population HUC guarantee for fixed-step two-stage HUC-Boost from a learned initializer)**.** *Run Algorithm 4 using the fixed step size with $\tau = 3\varepsilon/4$ and $\alpha_{\mathrm{fix}} = \alpha_\varepsilon$ under Assumption H.3, and let $0 < \delta < 1$. If*

$$n \geq \frac{512}{\varepsilon^2}\left[\Lambda_\varepsilon + (T_\varepsilon^{\mathrm{learn}} + 1)\log(2N_{\mathrm{cand}}(2n+1)^2) + \log\frac{1}{\delta}\right], \tag{166}$$

then, under the joint distribution of S and Ξ,

$$\mathbb{P}_{S,\Xi}(\mathrm{HUC}(\widehat{f}; \mathcal{C}, \mathcal{U}) \leq \varepsilon) \geq 1 - \delta. \tag{167}$$

Since $T_\varepsilon^{\mathrm{learn}} + 1 \leq 4D_0/\varepsilon^2 + 6$, condition Eq. (166) is implied by

$$n \geq \frac{512}{\varepsilon^2}\left[d \log\left(1 + \frac{2R_\theta L_\theta}{\varepsilon^2}\right) + \left(\frac{4D_0}{\varepsilon^2} + 6\right)\log(2N_{\mathrm{cand}}(2n+1)^2) + \log\frac{1}{\delta}\right].$$

When $D_0, R_\theta, L_\theta, d, N_{\mathrm{cand}}$ do not depend on ε, a sufficient condition with no n on the right-hand side is given in Appendix H.5, especially Corollary H.13. A sufficient sample-size order is

$$n = \widetilde{O}\left(\frac{D_0 + 1}{\varepsilon^4} + \frac{d + \log(1/\delta)}{\varepsilon^2}\right).$$

Thus, for fixed δ, learning the initial recalibration parameter from the same audit sample retains the same $n = \widetilde{O}(\varepsilon^{-4})$ dependence on accuracy as the fixed-initializer case. If pretraining uses independent data and its output satisfies Eq. (157) with D_{fix} independent of ε, conditioning on this output and applying Theorem H.2 gives the same $\widetilde{O}(\varepsilon^{-4})$ sufficient order.

H.3 Proofs of Theorems H.1 and H.2 for Fixed-Step HUC-Boost

We first prove the stopping theorem using the one-step empirical log-loss decrease in Appendix E.2.

Proof of Theorem H.1. Whenever the fixed-step specialization of Algorithm 3 updates, $|\widehat{\Gamma}_{c_t,u_t,I_t,v_t}(f_t)| > 3\varepsilon/4$. Substituting $\alpha = \alpha_\varepsilon = \varepsilon/2$ into Eq. (105) gives

$$\widehat{\mathcal{L}}_{\log,S}(f_{t+1}) < \widehat{\mathcal{L}}_{\log,S}(f_t) - \frac{\varepsilon^2}{4}.$$

After t updates,

$$0 \le \widehat{\mathcal{L}}_{\log,S}(f_t) < \widehat{\mathcal{L}}_{\log,S}(f_0) - \frac{t\varepsilon^2}{4}.$$

If updates continued until $t = \lceil 4\widehat{\mathcal{L}}_{\log,S}(f_0)/\varepsilon^2 \rceil$, the right-hand side would be nonpositive, a contradiction. Thus the update count satisfies Eq. (158). The stopping rule also gives $\widehat{\mathrm{HUC}}(\widehat{f};\mathcal{C},\mathcal{U}) \le 3\varepsilon/4$. □

Under Eq. (157), the stopping theorem shows that the output is reached in at most $T_\varepsilon^{\mathrm{fix}}$ updates and has empirical HUC at most $3\varepsilon/4$. It remains to control, uniformly over all predictors reachable in at most $T_\varepsilon^{\mathrm{fix}}$ adaptively selected fixed-step updates, the difference between population and empirical HUC by $\varepsilon/4$.

Definition H.6 (Reachable audit class and uniform generalization gap)**.** *Fix a positive initial predictor f_0 and an integer $T \ge 0$. Let $\mathcal{F}_{\le T}(f_0)$ be the set of all predictors obtainable from f_0 by applying at most T updates of Algorithm 3 in the fixed-step specialization, allowing arbitrary choices of (c,u,I,v) and update signs at each step. Define*

$$\mathcal{G}_{\le T}(f_0) := \Big\{ (x,y) \mapsto \varsigma c(x)\mathbf{1}\{\mu_u(f(x)) \in I\}\langle \boldsymbol{\Delta}_{u,v}(f(x)), \boldsymbol{R}_v(f(x),y)\rangle : \\ f \in \mathcal{F}_{\le T}(f_0),\ c \in \mathcal{C},\ u \in \mathcal{U},\ I \in \mathcal{I},\ v \in \mathcal{V}_{\mathcal{T}}(u),\ \varsigma \in \{-1,1\} \Big\} \quad (168)$$

and

$$\mathrm{Gap}_T(f_0,S) := \sup_{g \in \mathcal{G}_{\le T}(f_0)} \left\{ \mathbb{E}[g(X,Y)] - \frac{1}{n}\sum_{i=1}^n g(X_i,Y_i) \right\}. \quad (169)$$

Because ς is included, every $f \in \mathcal{F}_{\le T}(f_0)$ satisfies

$$\sup_{\substack{c\in\mathcal{C},\, u\in\mathcal{U},\, I\in\mathcal{I} \\ v\in\mathcal{V}_{\mathcal{T}}(u)}} |\Gamma_{c,u,I,v}(f) - \widehat{\Gamma}_{c,u,I,v}(f)| \le \mathrm{Gap}_T(f_0,S). \quad (170)$$

Consequently, by bounding the difference of two suprema by the supremum of the pointwise differences,

$$|\,\mathrm{HUC}(f;\mathcal{C},\mathcal{U}) - \widehat{\mathrm{HUC}}(f;\mathcal{C},\mathcal{U})| \le \mathrm{Gap}_T(f_0,S). \quad (171)$$

The remainder of the proof of Theorem H.2 has three steps. Step 1 counts the value vectors attained by $\mathcal{G}_{\le T}(f_0)$ on the finite sequence formed by the sample and an independent ghost sample. Step 2 uses this count and Rademacher signs to bound the exponential moment of the uniform generalization gap and derive a high-probability bound. Step 3 then uses the terminal empirical HUC.

Step 1: count reachable audit-value vectors on a finite sequence. After symmetrizing Eq. (169), it is enough to distinguish audit functions on the finite sequence consisting of the original and ghost samples. Since an audit value is determined by the final prediction, we first count the possible prediction-value vectors at each iteration.

Lemma H.7 (Reachable value vectors on a finite sequence)**.** *For every fixed positive initial predictor f_0, finite sequence $z_{1:m} = ((x_k,y_k))_{k=1}^m$, integer $T \ge 0$,*

$$|\{(f(x_k))_{k=1}^m : f \in \mathcal{F}_{\le T}(f_0)\}| \le \{2N_{\mathrm{cand}}(m+1)^2\}^T, \quad (172)$$

$$|\{(g(z_k))_{k=1}^m : g \in \mathcal{G}_{\le T}(f_0)\}| \le \{2N_{\mathrm{cand}}(m+1)^2\}^{T+1}. \quad (173)$$

Proof. Let $\mathcal{P}_t$ be the set of prediction-value vectors on $x_1, \dots, x_m$ obtainable after exactly t updates, and write $L_t := |\mathcal{P}_t|$. Since the initial predictor is fixed, $L_0 = 1$.

Fix a current vector $\boldsymbol{p} = (p_1, \dots, p_m) \in \mathcal{P}_t$. There are N_{cand} choices of (c, u, v) and two update signs. Once these are fixed, $c(x_k)$, $\mu_u(p_k)$, $\boldsymbol{\Delta}_{u,v}(p_k)$, and the current branch probabilities are determined at every k. The remaining information supplied by the interval I is only its membership set $\{k \in [m] : \mu_u(p_k) \in I\}$. Set $s_k := \mu_u(p_k)$ and order its distinct values as $r_1 < \cdots < r_q$. If an interval contains r_a and r_b with $a < \ell < b$, then $r_a < r_\ell < r_b$, so the interval property forces it to contain r_ℓ. Thus $I \cap \{r_1, \dots, r_q\}$ is either empty or a consecutive rank block $\{r_a, \dots, r_b\}$ for some $1 \le a \le b \le q$. Conversely, every such block is realized by the closed interval $[r_a, r_b]$. Interval membership therefore depends only on ranks, not on distances between scores, and observations with equal scores are always selected together. The nonempty blocks correspond to endpoint pairs (a, b) and number $q(q+1)/2$. Including the empty set, the number of membership sets is at most $1 + \frac{q(q+1)}{2} \le (q+1)^2 \le (m+1)^2$.

For a fixed current prediction vector, fixing (c, u, v), the sign, and the interval membership set uniquely determines the next prediction value at every x_k through the fixed-step branch of Algorithm 3. Hence

$$L_{t+1} \le 2N_{\mathrm{cand}}(m+1)^2 L_t. \tag{174}$$

Induction from $L_0 = 1$ gives $L_T \le \{2N_{\mathrm{cand}}(m+1)^2\}^T$. On the chosen finite sequence, an interval with an empty membership set leaves all prediction values unchanged. Thus every value vector reached within T steps on these points is represented by a length-T sequence, proving Eq. (172).

Once the final prediction vector is fixed, a signed audit-value vector is determined by a choice of (c, u, v), an interval membership set, and ς. Counting these choices gives Eq. (173), with at most $2N_{\mathrm{cand}}(m+1)^2$ possibilities. □

Step 2: symmetrize the uniform generalization gap.

Lemma H.8 (Exponential moment for a fixed initializer)**.** *For every fixed positive initial predictor f_0, integer $T \ge 0$, and $\lambda > 0$,*

$$\mathbb{E}_{S\sim\mathbb{P}^n}\left[\exp(\lambda \operatorname{Gap}_T(f_0, S))\right] \le \{2N_{\mathrm{cand}}(2n+1)^2\}^{T+1} \exp\left(\frac{8\lambda^2}{n}\right). \tag{175}$$

Proof. Write $\mathcal{G} := \mathcal{G}_{\le T}(f_0)$, which is independent of S. Let $S' = (Z'_1, \dots, Z'_n) \sim \mathbb{P}^n$ be an independent ghost sample, and let $\xi_1, \dots, \xi_n$ be independent Rademacher signs, independent of S, S'. Conditional Jensen's inequality and exchangeability of each pair (Z_i, Z'_i) give

$$\mathbb{E}_S e^{\lambda \operatorname{Gap}_T(f_0,S)} \le \mathbb{E}_{S,S',\xi} \exp\left(\frac{\lambda}{n} \sup_{g\in\mathcal{G}} \sum_{i=1}^n \xi_i [g(Z'_i) - g(Z_i)]\right).$$

Conditional on S, S', Lemma H.7 reduces this supremum to $L \le \{2N_{\mathrm{cand}}(2n+1)^2\}^{T+1}$ value vectors. Choose representatives $g_1, \dots, g_L$ and set $b_{j,i} := g_j(Z'_i) - g_j(Z_i)$; then $|b_{j,i}| \le 4$ because $|g_j| \le 2$. Hoeffding's lemma and $e^{\max_j a_j} \le \sum_j e^{a_j}$ yield

$$\mathbb{E}_\xi \exp\left(\frac{\lambda}{n} \max_{j\in[L]} \sum_{i=1}^n \xi_i b_{j,i}\right) \le \sum_{j=1}^L \exp\left(\frac{\lambda^2}{2n^2} \sum_{i=1}^n b_{j,i}^2\right) \le L e^{8\lambda^2/n}.$$

Taking expectations over S, S' proves Eq. (175). □

Lemma H.9 (Uniform generalization gap for a fixed initializer)**.** *For every fixed positive initial predictor f_0, integer $T \ge 0$, and $0 < \delta < 1$, with probability at least $1 - \delta$,*

$$\operatorname{Gap}_T(f_0, S) \le 4\sqrt{\frac{2}{n}\left[(T+1)\log(2N_{\mathrm{cand}}(2n+1)^2) + \log\frac{1}{\delta}\right]}. \tag{176}$$

Proof. Apply Markov's inequality to Lemma H.8 with $\lambda = n\rho/16$. For every $\rho > 0$,

$$\mathbb{P}(\operatorname{Gap}_T(f_0, S) > \rho) \le \exp\left((T+1)\log(2N_{\mathrm{cand}}(2n+1)^2) - \frac{n\rho^2}{32}\right).$$

Setting ρ to the right-hand side of Eq. (176) makes this probability at most δ. □

Step 3: combine empirical HUC at termination with the generalization gap.

Proof of Theorem H.2. Theorem H.1 and Eq. (157) give $\widehat{f} \in \mathcal{F}_{\le T_\varepsilon^{\mathrm{fix}}}(f_0)$. They also give $\widehat{\mathrm{HUC}}(\widehat{f};\mathcal{C},\mathcal{U}) \le 3\varepsilon/4$. Apply Lemma H.9 with $T = T_\varepsilon^{\mathrm{fix}}$. Under Eq. (160), its right-hand side is at most $\varepsilon/4$, and therefore, with probability at least $1-\delta$,

$$\mathrm{HUC}(\widehat{f};\mathcal{C},\mathcal{U}) \le \widehat{\mathrm{HUC}}(\widehat{f};\mathcal{C},\mathcal{U}) + \mathrm{Gap}_{T_\varepsilon^{\mathrm{fix}}}(f_0,S) \le \varepsilon.$$

□

H.4 Proofs for the Fixed-Step Two-Stage Method

Proof strategy. For the stopping count, we bound the initial empirical log loss after perturbation and apply Theorem H.1. For the population HUC guarantee, we first upper-bound the exponential moment of the uniform generalization gap for the sample-dependent initializer Θ_0 by an exponential moment under a sample-independent reference distribution. We then integrate the fixed-initializer bound of Lemma H.8 against this reference distribution, obtain a high-probability uniform generalization bound, and combine it with the empirical HUC at termination.

Step 1: initial loss after perturbation and the stopping count.

Lemma H.10 (Initial log loss after perturbation)**.** *Under Assumption H.3, almost surely,*

$$\widehat{\mathcal{L}}_{\log,S}(f_{0,\Theta_0}) \le D_0 + \varepsilon^2. \tag{177}$$

Proof. For $z, z' \in \mathbb{R}^K$, log-sum-exp is 1-Lipschitz with respect to $\|\cdot\|_\infty$. Hence the one-observation log loss $\ell_{\log}(z,y) = \log\sum_k e^{z_k} - z_y$ satisfies

$$|\ell_{\log}(z,y) - \ell_{\log}(z',y)| \le 2\|z - z'\|_\infty. \tag{178}$$

Since $\|\Xi\|_\infty \le a_\varepsilon$ and Eq. (163) holds, for every i,

$$\ell_{\log}(s_{\Theta_0}(X_i),Y_i) \le \ell_{\log}(s_{\widehat{\theta}}(X_i),Y_i) + 2L_\theta a_\varepsilon.$$

Averaging over i and using Eq. (162) and $2L_\theta a_\varepsilon = \varepsilon^2$ proves the claim. □

Proof of Theorem H.4. Fix a realization of (S,Ξ). Apply Theorem H.1 to this realized sample and the initial predictor f_{0,Θ_0}. Lemma H.10 gives

$$N_{\mathrm{upd}} \le \left\lceil \frac{4(D_0+\varepsilon^2)}{\varepsilon^2} \right\rceil = \left\lceil \frac{4D_0}{\varepsilon^2} \right\rceil + 4 = T_\varepsilon^{\mathrm{learn}}.$$

The stopping rule also gives $\widehat{\mathrm{HUC}}(\widehat{f};\mathcal{C},\mathcal{U}) \le 3\varepsilon/4$. □

Step 2: dominate the distribution of the sample-dependent initializer by a reference distribution. For Theorem H.5, with fixed $T \ge 0$ and $\lambda > 0$, the quantity to be bounded is

$$\mathbb{E}_{S,\Xi}\left[\exp\left(\lambda\,\mathrm{Gap}_T(f_{0,\Theta_0},S)\right)\right]. \tag{179}$$

Lemma H.8 controls this exponential moment for each fixed parameter θ. Introduce the reference distribution

$$\Pi_\varepsilon := \mathrm{Unif}\left([-R_\theta - a_\varepsilon, R_\theta + a_\varepsilon]^d\right), \tag{180}$$

and dominate the joint distribution of (S,Θ_0) by the product distribution $\mathbb{P}^n \otimes \Pi_\varepsilon$.

Lemma H.11 (Transfer from a sample-dependent initializer to the reference distribution)**.** *For every nonnegative measurable function* $\Phi : (\mathcal{X}\times\mathcal{Y})^n \times \mathbb{R}^d \to [0,\infty]$,

$$\mathbb{E}_{S,\Xi}[\Phi(S,\Theta_0)] \le e^{\Lambda_\varepsilon} \int_{\mathbb{R}^d} \mathbb{E}_{S\sim\mathbb{P}^n}[\Phi(S,\theta)]\,\Pi_\varepsilon(d\theta). \tag{181}$$

Proof. Fix a sample value $\mathbf{z}$ satisfying Eq. (162), which holds for $\mathbb{P}^n$-almost every $\mathbf{z}$, and write $\widehat{\theta}_{\mathbf{z}} := \mathsf{A}(\mathbf{z})$. Conditional on $S = \mathbf{z}$, the Lebesgue density of $\Theta_0 = \widehat{\theta}_{\mathbf{z}} + \Xi$ is

$$k_{\mathbf{z}}(\theta) = (2a_\varepsilon)^{-d} \mathbf{1}\left\{\theta \in \widehat{\theta}_{\mathbf{z}} + [-a_\varepsilon, a_\varepsilon]^d\right\}.$$

Because $\|\widehat{\theta}_{\mathbf{z}}\|_\infty \le R_\theta$, its support is contained in the support of Π_ε, whose Lebesgue density is

$$\pi_\varepsilon(\theta) = [2(R_\theta + a_\varepsilon)]^{-d} \mathbf{1}\left\{\theta \in [-R_\theta - a_\varepsilon, R_\theta + a_\varepsilon]^d\right\}.$$

Therefore, for Lebesgue-almost every θ,

$$k_{\mathbf{z}}(\theta) \le \left(1 + \frac{R_\theta}{a_\varepsilon}\right)^d \pi_\varepsilon(\theta) = e^{\Lambda_\varepsilon} \pi_\varepsilon(\theta). \tag{182}$$

Tonelli's theorem and Eq. (182) give

$$\mathbb{E}_{S,\Xi}[\Phi(S, \Theta_0)] = \mathbb{E}_S\left[\int_{\mathbb{R}^d} \Phi(S,\theta) k_S(\theta)\, d\theta\right] \le e^{\Lambda_\varepsilon} \int_{\mathbb{R}^d} \mathbb{E}_{S\sim\mathbb{P}^n}[\Phi(S,\theta)]\, \Pi_\varepsilon(d\theta). \qquad \square$$

Applying Eq. (181) with $\Phi(S,\theta) = \exp(\lambda \operatorname{Gap}_T(f_{0,\theta}, S))$ bounds Eq. (179) by the Π_ε-average of the fixed-θ exponential moments, multiplied by e^{Λ_ε}.

Step 3: a high-probability bound for the uniform generalization gap.

Lemma H.12 (Uniform generalization gap for a learned initializer)**.** *Under Assumption H.3, for every fixed integer $T \ge 0$ and $0 < \delta < 1$, with probability at least $1 - \delta$ under the joint distribution of S and Ξ,*

$$\operatorname{Gap}_T(f_{0,\Theta_0}, S) \le 4\sqrt{\frac{2}{n}\left[\Lambda_\varepsilon + (T+1)\log(2N_{\mathrm{cand}}(2n+1)^2) + \log\frac{1}{\delta}\right]}. \tag{183}$$

Proof. Apply Lemma H.11 with $\Phi(S,\theta) = e^{\lambda \operatorname{Gap}_T(f_{0,\theta},S)}$. Lemma H.8 gives the same bound for every fixed θ, so for every $\lambda > 0$,

$$\begin{aligned}\mathbb{E}_{S,\Xi} e^{\lambda \operatorname{Gap}_T(f_{0,\Theta_0},S)} &\le e^{\Lambda_\varepsilon} \int_{\mathbb{R}^d} \mathbb{E}_S e^{\lambda \operatorname{Gap}_T(f_{0,\theta},S)}\, \Pi_\varepsilon(d\theta) \\ &\le \exp\left(\Lambda_\varepsilon + (T+1)\log(2N_{\mathrm{cand}}(2n+1)^2) + \frac{8\lambda^2}{n}\right).\end{aligned} \tag{184}$$

Markov's inequality with $\lambda = n\rho/16$ gives, for every $\rho > 0$,

$$\mathbb{P}_{S,\Xi}(\operatorname{Gap}_T(f_{0,\Theta_0}, S) > \rho) \le \exp\left(\Lambda_\varepsilon + (T+1)\log(2N_{\mathrm{cand}}(2n+1)^2) - \frac{n\rho^2}{32}\right).$$

Setting ρ to the right-hand side of Eq. (183) proves the result. $\square$

Step 4: combine empirical HUC at termination with the generalization gap.

Proof of Theorem H.5. Theorem H.4 gives, almost surely, $\widehat{f} \in \mathcal{F}_{\le T_\varepsilon^{\mathrm{learn}}}(f_{0,\Theta_0})$ and $\widehat{\mathrm{HUC}}(\widehat{f}; \mathcal{C}, \mathcal{U}) \le 3\varepsilon/4$. Apply Lemma H.12 with $T = T_\varepsilon^{\mathrm{learn}}$. Under Eq. (166), the right-hand side of Eq. (183) is at most $\varepsilon/4$. Therefore, with probability at least $1 - \delta$,

$$\mathrm{HUC}(\widehat{f}; \mathcal{C}, \mathcal{U}) \le \widehat{\mathrm{HUC}}(\widehat{f}; \mathcal{C}, \mathcal{U}) + \operatorname{Gap}_{T_\varepsilon^{\mathrm{learn}}}(f_{0,\Theta_0}, S) \le \varepsilon.$$

$\square$

H.5 Explicite sample complexity analysis

The sample conditions in Theorems H.2 and H.5 both have the form

$$n \geq \frac{512}{\varepsilon^2}\left[\Lambda + (T+1)\log(2N_{\mathrm{cand}}(2n+1)^2) + \log\frac{1}{\delta}\right]. \tag{185}$$

For the fixed initializer, $(T, \Lambda) = (T_\varepsilon^{\mathrm{fix}}, 0)$; for the learned initializer, $(T, \Lambda) = (T_\varepsilon^{\mathrm{learn}}, \Lambda_\varepsilon)$.

Corollary H.13 (Explicit condition with n removed from the right-hand side)**.** *Let $\nu := T + 1$ and define*

$$n_\star(T, \Lambda, \varepsilon, \delta) := \left\lceil \frac{1024}{\varepsilon^2}\left[\Lambda + \nu\log(18N_{\mathrm{cand}}) + \log\frac{1}{\delta} + 2\nu\log\left(\frac{2048\nu}{\varepsilon^2}\right)\right]\right\rceil. \tag{186}$$

Every integer $n \geq n_\star(T, \Lambda, \varepsilon, \delta)$ satisfies Eq. (185)*.*

Proof. For $n \geq 1$, $2N_{\mathrm{cand}}(2n+1)^2 \leq 18N_{\mathrm{cand}}n^2$. It is therefore sufficient that $n \geq A + C\log n$, where

$$A := \frac{512}{\varepsilon^2}\left[\Lambda + \nu\log(18N_{\mathrm{cand}}) + \log(1/\delta)\right], \qquad C := \frac{1024\nu}{\varepsilon^2}.$$

Applying $\log t \leq t - 1$ with $t = n/(2C)$ gives $A + C\log n \leq A + n/2 + C\log(2C) - C$. Hence $n \geq 2A + 2C\log(2C)$ is sufficient, and substitution yields Eq. (186). □

H.6 Dirichlet, Temperature, and Vector Initial Models

For a fixed base prediction $q : \mathcal{X} \to \Delta^{K-1}$, consider Dirichlet calibration (Kull et al., 2019),

$$f_{0,W,b}(x) = \mathrm{softmax}(W\log q(x) + b), \qquad W \in \mathbb{R}^{K\times K}, \quad b \in \mathbb{R}^K, \tag{187}$$

where the logarithm is applied coordinatewise. As in the experimental representation, all K rows are treated as parameters.

Proposition H.14 (Constants for a Dirichlet initial model)**.** *Fix $q_{\min} \in (0, 1/K]$ and $\lambda_{\mathrm{reg}} > 0$ independently of S, and assume $q_y(x) \geq q_{\min}$ for every x, y. Let $\theta = \mathrm{vec}(W, b) \in \mathbb{R}^{K(K+1)}$, and suppose the initial fit satisfies*

$$\widehat{\mathcal{L}}_{\log,S}(f_{0,\widehat{\theta}}) + \frac{\lambda_{\mathrm{reg}}}{2}\|\widehat{\theta}\|_2^2 \leq \log K. \tag{188}$$

Then Assumption H.3 holds with

$$d = K(K+1), \qquad D_0 = \log K, \qquad R_\theta = \sqrt{\frac{2\log K}{\lambda_{\mathrm{reg}}}}, \qquad L_\theta = 1 + K\log(1/q_{\min}). \tag{189}$$

Proof. Let $\varphi(x) := (\log q_1(x), \ldots, \log q_K(x), 1)$, so $s_{\theta,y}(x) = \theta_y^\top\varphi(x)$. Since $0 < q_y(x) \leq 1$, $\|\varphi(x)\|_1 = 1 + \sum_{y=1}^K |\log q_y(x)| \leq 1 + K\log(1/q_{\min}) = L_\theta$. Consequently, $|s_{\theta,y}(x) - s_{\theta',y}(x)| \leq L_\theta\|\theta - \theta'\|_\infty$, which proves Eq. (163). Nonnegativity of log loss and Eq. (188) give $\widehat{\mathcal{L}}_{\log,S}(f_{0,\widehat{\theta}}) \leq \log K$. They also give $\|\widehat{\theta}\|_\infty \leq \|\widehat{\theta}\|_2 \leq \sqrt{2\log K/\lambda_{\mathrm{reg}}}$, which is Eq. (162). □

The choice $W = 0, b = 0$ gives the uniform prediction and makes the left-hand side of Eq. (188) equal to $\log K$. Thus it is enough to return an initial fit whose objective is no worse than this fixed candidate. If the base prediction has no positive lower bound, use, for a predetermined $\kappa_0 \in (0, 1)$, $q_y^{\kappa_0}(x) := (1 - \kappa_0)q_y(x) + \kappa_0/K$ as the input to the initial model; then $q_{\min} = \kappa_0/K$ is valid.

Corollary H.15 (Noise width and sample condition for a Dirichlet initial model)**.** *Under Proposition H.14, choose the uniform-noise width*

$$a_\varepsilon = \frac{\varepsilon^2}{2\{1 + K\log(1/q_{\min})\}}. \tag{190}$$

Then

$$T_\varepsilon^{\mathrm{learn}} = \left\lceil \frac{4\log K}{\varepsilon^2} \right\rceil + 4, \qquad \Lambda_\varepsilon = K(K+1)\log\left(1 + \frac{2R_\theta L_\theta}{\varepsilon^2}\right). \tag{191}$$

Under the sample condition obtained by substituting these constants into Eq. (186)*,* $\mathrm{HUC}(\widehat{f}; \mathcal{C}, \mathcal{U}) \le \varepsilon$ *with probability at least* $1-\delta$*. The dependence on accuracy is*

$$n_\star = O\left(\frac{1+\log K}{\varepsilon^4}\log\frac{eN_{\mathrm{cand}}(1+\log K)}{\varepsilon^4} + \frac{K(K+1)}{\varepsilon^2}\log\left(1+\frac{2R_\theta L_\theta}{\varepsilon^2}\right) + \frac{\log(1/\delta)}{\varepsilon^2}\right). \tag{192}$$

Proof. Substitute the constants from Proposition H.14 into Theorem H.5 and Corollary H.13. Eq. (192) follows from $T_\varepsilon^{\mathrm{learn}} + 1 = O((1+\log K)/\varepsilon^2)$ and Eq. (191). □

Subtracting the Kth logit from all logits leaves softmax unchanged. If initial fitting and perturbation are defined in coordinates with the Kth row fixed to zero, one may use $d = (K-1)(K+1) = K^2-1$. The same proof applies when Eq. (188) is imposed on all free coefficients in those coordinates.

Application to temperature scaling. Let $L_q := \log(1/q_{\min})$. Temperature scaling, parameterized by the inverse temperature β, can be written as

$$s_{\beta,y}(x) = \beta \log q_y(x), \qquad d = 1, \qquad L_\theta = L_q. \tag{193}$$

Indeed, $\sup_x \|s_\beta(x) - s_{\beta'}(x)\|_\infty \le L_q|\beta-\beta'|$. Restricting the fitted parameter to a bounded interval determines R_θ, and D_0 may be obtained from an objective-comparison guarantee or a bound on the logits. To preserve a positive inverse temperature, choose the lower endpoint of the fitting domain so that $\beta_{\min} > a_\varepsilon$; then the perturbed parameter also remains positive.

Application to vector scaling. Let $\theta = (a, b) \in \mathbb{R}^{2K}$ and write vector scaling as

$$s_{\theta,y}(x) = a_y \log q_y(x) + b_y, \qquad d = 2K, \qquad L_\theta = L_q + 1. \tag{194}$$

For any $\theta = (a,b)$ and $\theta' = (a',b')$, $\sup_x \|s_\theta(x) - s_{\theta'}(x)\|_\infty \le (L_q+1)\|\theta - \theta'\|_\infty$. Thus Assumption H.3 applies with $d = 2K$ and $L_\theta = L_q + 1$ whenever the fitted parameter is restricted to a bounded domain and the initial empirical log loss is bounded by D_0.

Remark H.16 (Role and Scope of the One-Time Random Perturbation). When the initial predictor is fixed independently of the audit sample, no random perturbation is needed. When $\widehat{\theta} = \mathsf{A}(S)$ is learned from the same sample, the perturbation transfers the fixed-initializer uniform generalization bound to the sample-dependent initializer, at the density cost Λ_ε.

The theorem guarantees the output obtained after adding the stated perturbation once and running the fixed step $\alpha_\varepsilon = \varepsilon/2$.

The width $a_\varepsilon = \varepsilon^2/(2L_\theta)$ balances the density ratio and the initial loss increase. For a general width $a > 0$, the log density ratio is $d\log(1 + R_\theta/a)$, whereas the increase in the initial empirical log loss is at most $2L_\theta a$. The selected width limits this increase to ε^2, adding four to the update bound.

I Additional Comparison with Related Work

U-calibration and unknown downstream utilities. Kleinberg et al. (2023) study sequential forecasts of binary outcomes used by agents whose utilities are unknown to the forecaster. Their U-calibration error is the worst-case regret over bounded proper scoring rules, measured against the best fixed forecast in hindsight. This connects forecast evaluation to regret guarantees for agents that best respond to the forecasts. U-calibration evaluates cumulative regret, whereas the Utility Calibration of Hegazy et al. (2026) audits expected utility residuals within predicted-utility intervals. HUC decomposes these residuals along a fixed label tree and audits the resulting node contributions for the specified utility class.

Loss outcome indistinguishability and omniprediction. Gopalan et al. (2023) compare actual outcomes with outcomes sampled from a predictor using statistical tests induced by a loss family and a comparison hypothesis class. Their loss outcome indistinguishability condition implies omniprediction: loss-specific post-processing competes with the best hypothesis in the class for each loss. This shares with HUC the use of expected discrepancies evaluated through a specified family of tests. The distinction lies in the tests and the resulting guarantees. HUC uses the node contributions of a fixed target utility under common subgroup weights and predicted-utility intervals. Its guarantees concern these calibration moments and the restricted monotone post-processing comparisons in Appendix G, rather than loss minimization against a general hypothesis class.

Utility-directed conformal prediction. Cortes-Gomez et al. (2025) incorporate downstream decision losses and user-specified utilities into the construction of conformal prediction sets while retaining marginal coverage guarantees. Their healthcare application also uses a hierarchy of dermatological diseases to produce diagnostically coherent sets. This shares our interest in utility-aware use of label hierarchies, but the statistical targets differ. Conformal coverage concerns whether the realized label belongs to the reported set. HUC instead audits the agreement between predicted and realized utility through internal-node contributions to a probability forecast's utility residual. Thus, utility-directed set construction and hierarchical utility calibration address different aspects of reliable decision-making.

J Toy Data

We first specify the single-sample eight-leaf correction experiment used in the main text. A separate subsection explains the decision-utility construction.

J.1 Experimental Settings of the Main Experiment

Inputs, subgroups, and utilities. The model has eight leaf labels and seven internal nodes. Its initial prediction is specified directly, without fitting a classifier to input features. All evaluations use the whole-population indicator, $\mathcal{C} = \{1\}$, so $|\mathcal{C}| = 1$. The main experiment uses the three decision utilities defined below, with $|\mathcal{U}| = 3$.

Tree. Consider the binary tree in Figure 3, with eight leaf labels and seven internal nodes. Although synthetic, it can be interpreted as a staged classification of conditions in patients presenting with symptoms. First, the root v_{cause} distinguishes noninfectious from infectious causes. The noninfectious branch v_{noninf} then separates mild and severe conditions, y_1, y_2. For infectious causes, v_{spread} distinguishes localized from systemic infection. Within localized infection, v_{site} separates upper respiratory infection from pneumonia; v_{upper} and v_{lower} then distinguish viral from bacterial causes, giving y_3, y_4 and y_5, y_6, respectively. Within systemic infection, v_{shock} separates nonsevere and severe conditions, y_7, y_8. These synthetic interpretations are not clinical recommendations.

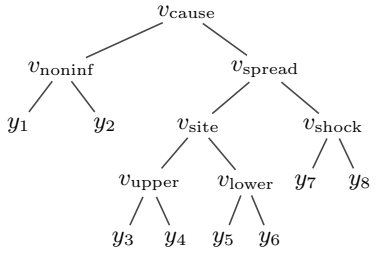


Figure 3: The eight-leaf binary tree used in the Toy experiments.

Decision utilities. Each utility evaluates the action selected from the prediction p against the realized label y. For decision j, let D_j be the leaves on which the decision is active, $S_j \subseteq D_j$ the leaves

favoring action 1 over action 0, and $\tau_j \in (0,1)$ its threshold. Define

$$d_j(y) = \mathbf{1}\{y \in S_j\} - \tau_j \mathbf{1}\{y \in D_j\}, \qquad \widehat{a}_j(p) = \mathbf{1}\left\{\sum_y p_y d_j(y) \geq 0\right\}, \tag{195}$$
$$u_j(p,y) = \{2\widehat{a}_j(p) - 1\} d_j(y).$$

The three decisions use

j	D_j	S_j	τ_j
1	$\{y_7, y_8\}$	$\{y_8\}$	0.40
2	$\{y_1, \ldots, y_8\}$	$\{y_2, y_5, y_6, y_7, y_8\}$	0.50
3	$\{y_1, \ldots, y_8\}$	$\{y_4, y_6, y_7, y_8\}$	0.40

(196)

Their relevant-node sets contain 3, 4, and 5 nodes, and their union covers all seven internal nodes.

Population distribution and random initial predictions. The true leaf distribution is

$$\eta = \left(\frac{13}{80}, \frac{7}{80}, \frac{21}{125}, \frac{21}{500}, \frac{189}{2000}, \frac{441}{2000}, \frac{9}{80}, \frac{9}{80}\right). \tag{197}$$

For each repetition, the predicted right-branch probabilities at the seven internal nodes are drawn independently from $\mathrm{Unif}(0.1, 0.9)$. Multiplying the sampled branch probabilities along each root-to-leaf path gives a positive initial leaf distribution p_0. Both corrections start from this p_0.

Calibration sample. For each sample size n, draw i.i.d. labels $Y_1, \ldots, Y_n$ from η and form

$$\widehat{\eta}_y = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\{Y_i = y\}, \qquad n\widehat{\eta} \sim \mathrm{Multinomial}(n, \eta). \tag{198}$$

Both corrections use this same calibration sample throughout their updates; n is the total number of calibration labels.

Correction procedures. At each UC-Boost iteration, choose the utility with the largest absolute empirical aggregate residual. For the selected utility u and current prediction p_t, set

$$h_{t,y} = u(p_t, y) - \mu_u(p_t), \qquad \widehat{\Gamma}_t = \sum_y \widehat{\eta}_y h_{t,y},$$
$$\widehat{\Lambda}_t = \frac{1}{4}\left(\max_y h_{t,y} - \min_y h_{t,y}\right)^2, \quad \alpha_t = \widehat{\Gamma}_t / \widehat{\Lambda}_t, \tag{199}$$

and update all leaf logits by

$$p_{t+1} = \mathrm{softmax}(\log p_t + \alpha_t h_t). \tag{200}$$

HUC-Boost instead selects the utility–node pair with the largest absolute empirical node contribution over all utilities and their relevant nodes. It applies Algorithm 1 with every expectation replaced by the average over the same calibration sample: the selected node's branch logits are updated with coefficient $\widehat{\Gamma}_t/\widehat{\Lambda}_t$, and the leaf distribution is reconstructed by path products. This preserves the selected node's reach probability, the conditional leaf distribution within each child subtree, and every leaf probability outside the selected subtree.

Both methods apply each selected update once and then recompute the utilities and all audit candidates at the new prediction. They stop when their respective empirical UC or HUC is at most 10^{-10}, or when the update cap is reached, and return the final prediction. The caps are 100 updates for UC-Boost and $100 \times 7 = 700$ updates for HUC-Boost. All runs, including those reaching a cap, are included in the reported results.

Exact population evaluation and repetitions. We evaluate the final corrected predictions by exact population UC and HUC under the known η. Thus the reported errors reflect finite-sample correction without an additional finite-test-sample error. We use

$$n \in \{250, 500, 1000, 2000, 5000, 10^4, 2\times10^4, 5\times10^4, 10^5, 2\times10^5, 5\times10^5, 10^6\}$$

and perform 200 repetitions per sample size, reusing the same 200 initial predictions across all sample sizes. Figure 2(b) reports the means, with one sample standard deviation at selected sizes.

J.2 Interpretation of the Decision Utilities

The construction in Eq. (195) evaluates an action selected from the prediction p against the realized label y. This subsection explains the roles of its outcome sets and threshold.

Outcome sets and label-wise values. For decision j, D_j is the set of outcomes on which the action comparison is evaluated. Within D_j, the outcomes in S_j favor action 1, whereas those in $D_j \setminus S_j$ favor action 0. Outside D_j, both actions receive zero utility. For $0 < \tau_j < 1$, the definition of d_j gives

$$d_j(y) = \begin{cases} 1-\tau_j, & y \in S_j, \\ -\tau_j, & y \in D_j \setminus S_j, \\ 0, & y \notin D_j. \end{cases}$$

Thus $d_j(y)$ is positive on outcomes favoring action 1 and negative on outcomes favoring action 0.

Payoffs of the two actions. Write the action-level payoff as $U_j(a, y) := (2a-1)d_j(y), a \in \{0, 1\}$. Action 1 receives $d_j(y)$, and action 0 receives $-d_j(y)$:

Realized outcome	$U_j(1, y)$	$U_j(0, y)$
$y \in S_j$	$1-\tau_j$	$-(1-\tau_j)$
$y \in D_j \setminus S_j$	$-\tau_j$	τ_j
$y \notin D_j$	0	0

On D_j, choosing the preferred action gives positive utility, and choosing the other action gives negative utility. The parameter τ_j controls the relative magnitudes of these payoffs: choosing action 0 on S_j gives $-(1-\tau_j)$, whereas choosing action 1 on $D_j \setminus S_j$ gives $-\tau_j$. These two penalties have equal magnitude when $\tau_j = 1/2$.

Prediction-based choice and the threshold. For $A \subseteq \mathcal{Y}$, write $p(A) := \sum_{y \in A} p_y$. The predicted expected payoffs of actions 1 and 0 are, respectively,

$$\sum_y p_y U_j(1, y) = p(S_j) - \tau_j p(D_j), \qquad \sum_y p_y U_j(0, y) = -\{p(S_j) - \tau_j p(D_j)\}.$$

The rule $\widehat{a}_j(p)$ therefore selects the action with the larger predicted expected payoff, choosing action 1 in a tie. Since $p(D_j) > 0$ in this experiment, $\widehat{a}_j(p) = 1 \iff \frac{p(S_j)}{p(D_j)} \geq \tau_j$. Thus τ_j is the threshold for the predicted probability of an outcome favoring action 1 conditional on the outcome lying in D_j. When $D_j = \mathcal{Y}$, the condition reduces to $p(S_j) \geq \tau_j$. The prediction selects the action before the outcome is observed, and the realized label determines its payoff: $u_j(p, y) = U_j\big(\widehat{a}_j(p), y\big)$.

K Real Data

The exact interval maximization for empirical UC and HUC is described in Appendix F.3 (Algorithm 2).

We evaluate iNaturalist 2019, SUPPORT2, MASSIVE, NSL-KDD, and N-BaIoT. The first two provide the main-text comparisons; the others cover text and network-traffic classification.

K.1 Common Experimental Settings

Training and evaluation. Base predictors are fitted on training data, and recalibration uses separate calibration data. Validation data select regularization and the returned corrected predictor; test data are reserved for final evaluation. We evaluate UC and HUC, as well as accuracy and AUC. AUC is the unweighted average of one-vs-rest ROC AUC values over classes with both positive and negative examples in the evaluation set. Each dataset uses five data splits, with its sample sizes and allocation described below. Within each split, all calibration methods share the same base predictions. Results are reported as means and sample standard deviations. For the six-classifier comparisons, pooled results give equal weight to the six classifiers and five splits.

Base predictors. Except for MASSIVE, we compare logistic regression (LR), Gaussian naive Bayes (GNB), decision trees (DT), random forests (RF), gradient-boosted trees (XGB), and a multilayer perceptron (MLP). Each classifier predicts the leaf labels directly. The probability of an internal node is the sum of its descendant-leaf probabilities; no separate classifier is trained at an internal node. The input features and their dimensions are specified for each dataset below.

Calibration methods. Baseline denotes the common positive base prediction. For a fitted base predictor f_{raw}, we set

$$f_{\mathrm{base}}(x) = (1 - \kappa_{\mathrm{init}}) f_{\mathrm{raw}}(x) + \frac{\kappa_{\mathrm{init}}}{K}\mathbf{1}, \qquad \kappa_{\mathrm{init}} = 10^{-10}, \tag{201}$$

using the same coefficient for every real-data dataset and calibration method. This transformation is applied once to the base predictions before recalibration; it is not repeated after boosting updates. We compare direct UC-Boost and HUC-Boost, temperature scaling, vector scaling (Guo et al., 2017), and Dirichlet calibration (Kull et al., 2019). The three parametric transformations are defined in Table 4.

Table 4: Whole-distribution recalibration baselines and the parameters learned on calibration data.

Method	Transformation	Parameters learned on calibration data
Temperature	$g_{\mathrm{Temp},T}(p) = \mathrm{softmax}(\log p / T)$	scalar $T > 0$
Vector	$g_{\mathrm{Vec},a,b_{\mathrm{V}}}(p) = \mathrm{softmax}(a \odot \log p + b_{\mathrm{V}})$	$a, b_{\mathrm{V}} \in \mathbb{R}^K$
Dirichlet	$g_{\mathrm{Dir},W,b_{\mathrm{D}}}(p) = \mathrm{softmax}(W \log p + b_{\mathrm{D}})$	$W \in \mathbb{R}^{K\times K}$, $b_{\mathrm{D}} \in \mathbb{R}^K$

For $p = f_{\mathrm{base}}(x)$, the parameters of each transformation are fitted by minimizing calibration negative log likelihood,

$$\widehat{\mathcal{L}}_{\mathrm{cal}}(\theta_g) = -\frac{1}{|\mathcal{D}_{\mathrm{cal}}|} \sum_{(x_i,y_i)\in\mathcal{D}_{\mathrm{cal}}} \log\left(\left[g_{\theta_g}(f_{\mathrm{base}}(x_i))\right]_{y_i}\right). \tag{202}$$

Vector and Dirichlet calibration additionally use regularization, whose strength is selected on validation data. After fitting each parametric recalibrator, we apply the one-time independent uniform perturbation to its free parameters described in Appendix H, using perturbation parameter $\varepsilon_{\mathrm{noise}} = 10^{-3}$ and master random seed 20260907. Let $\widetilde{\theta}_g$ denote the resulting perturbed parameter. The same perturbed map $g_{\widetilde{\theta}_g}$ is used for the standalone parametric method and as the initializer of the corresponding two-stage HUC-Boost procedure. We therefore start HUC-Boost from

$$f_g^{(0)}(x) = g_{\widetilde{\theta}_g}(f_{\mathrm{base}}(x)). \tag{203}$$

An arrow denotes this two-stage composition. HUC-Boost uses the step $\alpha = \Gamma/\Lambda$. HUC-Boost (0.05) denotes the comparison with constant step magnitude 0.05.

Iteration budgets. UC-Boost uses at most 100 updates. Each HUC-Boost procedure, including the fixed-step and two-stage variants, uses at most 100 passes over the nodes with at least two children, for a node-audit budget of 100 times the number of such nodes. A node is updated only when its audited violation exceeds 0.001; one-child nodes are excluded. UC-Boost stops when its empirical violation is at most 0.001, and HUC-Boost stops when a complete pass finds no violation, or when the respective budget is exhausted. Validation selects the returned update prefix.

Utilities, subgroups, and reported quantities. Each dataset specifies its target utilities $\mathcal{U}$ and subgroup family $\mathcal{C}$. For a fixed prediction p, its predicted mean utility is $\mu_u(p) = \sum_{y\in\mathcal{Y}} p_y u(p, y)$; any action or reported node selected from p is held fixed while averaging over the label y. Subgroups are represented by binary indicators $c_A(x) = \mathbf{1}\{x \in A\}$, including $c \equiv 1$. UC and HUC use the same utility and subgroup families within each comparison. We also report the number of updates retained in the selected predictor.

K.2 iNaturalist 2019 (Van Horn et al., 2018)

Data and label hierarchy. iNaturalist associates each image with a species label and a biological taxonomy. We use 3,344 images from 30 species, with at most 120 images per species. Each of five data splits contains 1,505 training, 669 calibration, 503 validation, and 667 test images. The species are the leaves of the supplied biological taxonomy, shown in Figure 4. Let V_ℓ denote its nodes at depth ℓ; depths $3, 4, 5, 6, 7$ correspond to class, order, family, genus, and species, respectively.

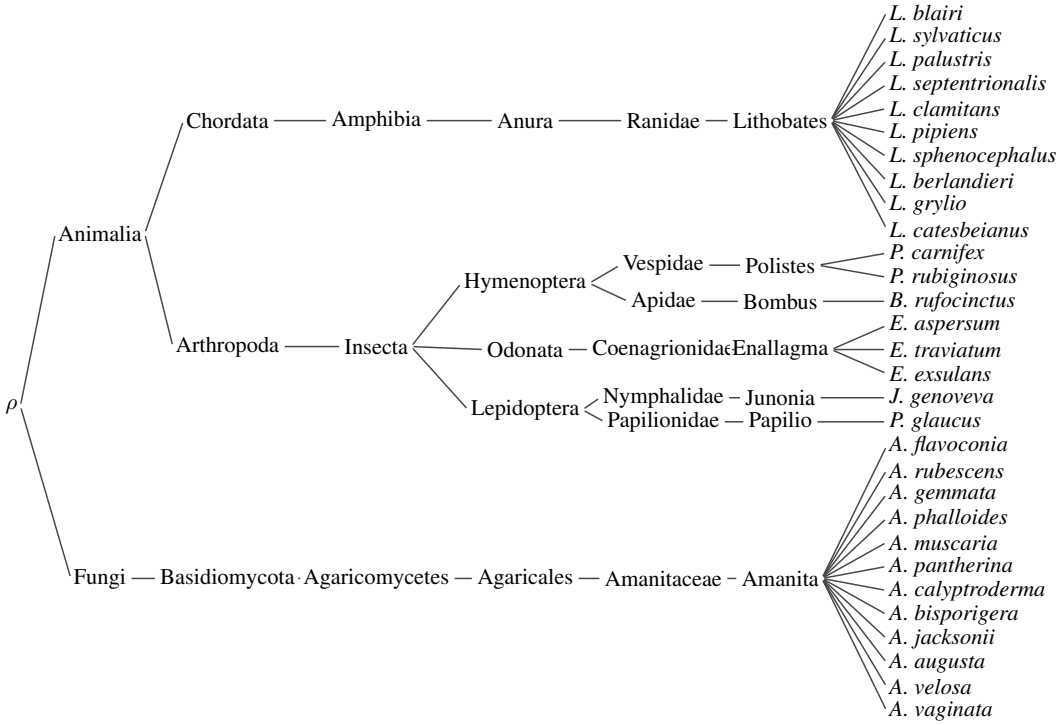


Figure 4: The biological label tree for the selected iNaturalist species. Species labels abbreviate the genus, which is written at their parent node. Every line denotes a parent–child edge.

Features. Each image is represented by the 512-dimensional feature vector before the final classification layer of an ImageNet-pretrained ResNet-18, which is not fine-tuned on iNaturalist. The six base predictors in Appendix K.1 are trained on these features to predict the 30 species.

Utilities. For each node v, let $\mathcal{Y}_v$ be the species below v and define $J_v(y) := \mathbf{1}\{y \in \mathcal{Y}_v\}$, $\pi_v(p) := \sum_{z \in \mathcal{Y}_v} p_z$. Thus $J_v(y)$ is one if the true species y belongs to the category represented by v, and zero otherwise; $\pi_v(p)$ is that category's predicted probability. At each level $\ell \in \{3, 4, 5, 6, 7\}$, select the single node with the largest predicted probability,

$$\widehat{v}_\ell(p) := \operatorname*{arg\,max}_{\prec;\, v \in V_\ell} \pi_v(p), \tag{204}$$

with ties resolved by the fixed rule in Appendix A.1. Its utility and predicted mean are

$$\begin{aligned} u_\ell(p, y) &:= J_{\widehat{v}_\ell(p)}(y) = \mathbf{1}\{y \in \mathcal{Y}_{\widehat{v}_\ell(p)}\}, \\ \mu_{u_\ell}(p) &= \sum_{z \in \mathcal{Y}} p_z \mathbf{1}\{z \in \mathcal{Y}_{\widehat{v}_\ell(p)}\} = \sum_{z \in \mathcal{Y}_{\widehat{v}_{\ell(p)}}} p_z = \pi_{\widehat{v}_\ell(p)}(p). \end{aligned} \tag{205}$$

The sets $\{\mathcal{Y}_v : v \in V_\ell\}$ partition the species, so each true species belongs to exactly one category at level ℓ. The indicator above is therefore one exactly when the most probable category $\widehat{v}_\ell(p)$ is the true category at that level. This is top-1 correctness at class, order, family, genus, and species for $\ell = 3, 4, 5, 6, 7$, respectively. At the species level, $\mathcal{Y}_v = \{v\}$, so the rule reduces to selecting the most

probable leaf and checking whether it equals y. At higher levels, selection uses the total probability of each category's descendant species. UC and HUC compare this realized 0–1 correctness with the model's probability that the selected category is correct.

Subgroups. The five utilities give $|\mathcal{U}| = 5$. There are seven subgroup indicators, $|\mathcal{C}| = 7$: the whole population; brightness below and at least its training median; contrast below and at least its training median; and aspect ratio below one and at least one. Aspect ratio is image width divided by height. Indicators use only these separate attributes, without intersections or other input features.

Results. Table 5 gives the pooled results for the five utilities. HUC-Boost gives lower mean HUC than the fixed-step correction. The fixed-step correction retains more updates and has higher mean UC and lower mean AUC.

Table 5: Complete iNaturalist results for the five rank-wise correctness utilities. Values are mean ± sample standard deviation over six base predictors and five splits. UC-Boost and HUC-Boost are abbreviated as UC-B and HUC-B, respectively.

Method	Acc. ↑	AUC ↑	UC ↓	HUC ↓	Updates
Baseline	.4886±.0895	.9001±.1082	.2721±.1615	.1251±.0595	0.00±0.00
UC-B	.4618±.0877	.9138±.0664	**.0452±.0177**	.0361±.0108	25.43±24.07
HUC-B	.4633±.0896	.9223±.0599	.0748±.0317	.0345±.0090	145.93±180.47
HUC-B (0.05)	.4853±.0888	.8971±.1106	.1837±.1767	.0815±.0725	329.23±270.41
Temp.	.4886±.0895	.8902±.1097	.1303±.1143	.0920±.0756	0.00±0.00
Temp.→HUC	.4702±.0878	**.9301±.0501**	.0473±.0135	**.0289±.0081**	43.07±25.92
Vec.	**.4888±.0952**	.8986±.0999	.1144±.0922	.0827±.0726	0.00±0.00
Vec.→HUC	.4781±.0950	.9291±.0528	.0496±.0176	.0313±.0079	28.23±22.38
Dir.	.4365±.1125	.9119±.0626	.1047±.0532	.0551±.0230	0.00±0.00
Dir.→HUC	.4349±.1116	.9120±.0623	.1093±.0523	.0556±.0227	4.27±10.53

Results by base predictor. Tables 6 and 7 separate all procedures by base predictor. HUC-Boost gives a much smaller HUC error than HUC-Boost (0.05) for LR, GNB, and DT, which are the principal sources of the pooled difference. The fixed-step method uses more updates for most base predictors; DT is the exception because its fixed path stops with far fewer selected updates but also has much larger HUC. RF and XGB provide small exceptions in which the fixed step has a slightly smaller HUC, so the aggregate conclusion does not imply the same ordering for every base predictor.

Table 6: iNaturalist (all utilities): baseline methods by base predictor. Values are mean ± sample standard deviation over five data splits. UC-Boost and HUC-Boost are abbreviated as UC-B and HUC-B, respectively.

	LR					**RF**				
Method	Acc. ↑	AUC ↑	UC ↓	HUC ↓	Updates	Acc. ↑	AUC ↑	UC ↓	HUC ↓	Updates
Baseline	.5436±.0222	.9522±.0028	.2439±.0136	.1225±.0066	0.00±0.00	.5187±.0081	.9509±.0027	.2957±.0089	.1246±.0089	0.00±0.00
UC-B	.4786±.0206	.9531±.0031	.0721±.0135	.0536±.0043	9.80±3.42	.5178±.0069	.9550±.0027	.0299±.0037	.0328±.0067	18.80±7.46
Temp.	.5436±.0222	.9537±.0030	.0704±.0072	.0574±.0073	0.00±0.00	.5187±.0081	.9572±.0022	**.0285±.0043**	.0249±.0049	0.00±0.00
Vec.	**.5460±.0192**	**.9556±.0028**	**.0639±.0082**	**.0379±.0060**	0.00±0.00	**.5220±.0155**	**.9573±.0021**	.0390±.0184	**.0215±.0053**	0.00±0.00
Dir.	.5055±.0115	.9521±.0028	.1810±.0101	.0891±.0098	0.00±0.00	.2984±.0157	.8933±.0067	.1638±.0076	.0769±.0067	0.00±0.00
	GNB					**XGB**				
Method	Acc. ↑	AUC ↑	UC ↓	HUC ↓	Updates	Acc. ↑	AUC ↑	UC ↓	HUC ↓	Updates
Baseline	.5244±.0213	.9245±.0107	.4512±.0212	.1977±.0164	0.00±0.00	.5145±.0148	.9526±.0031	.0778±.0105	.0525±.0061	0.00±0.00
UC-B	.4774±.0181	.8791±.0114	**.0533±.0098**	**.0400±.0119**	34.60±2.07	.5040±.0090	.9559±.0022	.0420±.0125	.0298±.0050	11.20±4.76
Temp.	.5244±.0213	.8582±.0161	.2089±.0098	.1426±.0087	0.00±0.00	.5145±.0148	.9525±.0031	.0814±.0125	.0526±.0078	0.00±0.00
Vec.	.5253±.0161	.8807±.0132	.1787±.0120	.1355±.0090	0.00±0.00	**.5178±.0118**	.9526±.0034	.0732±.0073	.0494±.0078	0.00±0.00
Dir.	**.5277±.0184**	**.9298±.0080**	.0754±.0144	.0444±.0097	0.00±0.00	.5127±.0090	**.9573±.0032**	**.0417±.0051**	**.0269±.0053**	0.00±0.00
	DT					**MLP**				
Method	Acc. ↑	AUC ↑	UC ↓	HUC ↓	Updates	Acc. ↑	AUC ↑	UC ↓	HUC ↓	Updates
Baseline	**.2960±.0181**	.6637±.0096	.4778±.0188	.1948±.0110	0.00±0.00	.5343±.0142	.9565±.0026	.0860±.0513	.0587±.0176	0.00±0.00
UC-B	.2750±.0167	.7825±.0115	**.0437±.0152**	**.0337±.0049**	70.60±20.31	.5178±.0143	.9571±.0033	**.0304±.0051**	**.0267±.0022**	7.60±5.22
Temp.	**.2960±.0181**	.6629±.0096	.3446±.0140	.2348±.0075	0.00±0.00	.5343±.0142	.9567±.0027	.0478±.0077	.0397±.0044	0.00±0.00
Vec.	.2828±.0107	.6883±.0089	.2865±.0163	.2175±.0166	0.00±0.00	**.5388±.0114**	**.9572±.0026**	.0451±.0093	.0342±.0070	0.00±0.00
Dir.	.2666±.0280	**.7840±.0082**	.0971±.0119	.0481±.0109	0.00±0.00	.5079±.0140	.9549±.0022	.0695±.0267	.0455±.0090	0.00±0.00

Table 7: iNaturalist (all utilities): HUC-Boost results by base predictor. Values are mean ± sample standard deviation over five data splits. UC-Boost and HUC-Boost are abbreviated as UC-B and HUC-B, respectively.

	LR					RF				
Method	Acc. ↑	AUC ↑	UC ↓	HUC ↓	Updates	Acc. ↑	AUC ↑	UC ↓	HUC ↓	Updates
HUC-B	.5094±.0049	.9550±.0027	.0876±.0050	.0354±.0048	41.80±7.09	.5034±.0094	.9513±.0030	.0429±.0116	.0286±.0034	79.20±21.49
HUC-B (0.05)	.5286±.0144	.9559±.0031	.1124±.0198	.0446±.0081	400.20±62.97	.5196±.0063	.9544±.0021	.0410±.0081	.0206±.0033	385.20±70.12
Temp.→HUC	.5181±.0093	.9560±.0025	.0584±.0114	**.0336±.0090**	46.00±31.02	.5070±.0116	.9564±.0028	.0315±.0072	**.0204±.0007**	29.60±22.13
Vec.→HUC	**.5406±.0104**	**.9566±.0028**	**.0551±.0109**	.0349±.0055	17.80±14.86	**.5226±.0122**	**.9571±.0022**	**.0298±.0024**	.0232±.0073	12.00±12.90
Dir.→HUC	.5055±.0115	.9521±.0028	.1810±.0101	.0891±.0098	0.00±0.00	.2984±.0157	.8933±.0067	.1638±.0076	.0769±.0067	0.00±0.00
	GNB					**XGB**				
Method	Acc. ↑	AUC ↑	UC ↓	HUC ↓	Updates	Acc. ↑	AUC ↑	UC ↓	HUC ↓	Updates
HUC-B	.4894±.0244	.9225±.0092	.1108±.0140	.0426±.0074	190.00±23.82	.4948±.0133	.9545±.0026	.0586±.0089	.0360±.0038	30.80±13.88
HUC-B (0.05)	**.5280±.0228**	.8986±.0262	.3707±.0226	.1644±.0181	751.00±57.04	.5106±.0210	.9557±.0025	.0528±.0138	.0319±.0067	152.20±72.97
Temp.→HUC	.5013±.0223	**.9349±.0069**	**.0598±.0080**	**.0345±.0086**	58.80±18.36	.4978±.0124	.9545±.0027	.0563±.0045	.0338±.0076	34.60±11.74
Vec.→HUC	.5124±.0309	.9345±.0071	.0660±.0241	.0362±.0071	43.40±13.03	.4939±.0109	.9544±.0022	.0577±.0163	.0378±.0057	31.80±8.76
Dir.→HUC	.5190±.0232	.9298±.0082	.0928±.0185	.0439±.0051	14.80±11.86	**.5127±.0090**	**.9573±.0032**	**.0417±.0051**	**.0269±.0053**	0.00±0.00
	DT					**MLP**				
Method	Acc. ↑	AUC ↑	UC ↓	HUC ↓	Updates	Acc. ↑	AUC ↑	UC ↓	HUC ↓	Updates
HUC-B	.2699±.0158	.7936±.0047	.1006±.0236	.0392±.0060	507.00±126.11	.5127±.0225	.9567±.0022	.0484±.0339	.0252±.0136	26.80±11.90
HUC-B (0.05)	**.2939±.0210**	.6599±.0163	.4741±.0119	.1955±.0123	172.80±386.39	**.5310±.0130**	**.9581±.0024**	.0514±.0268	.0319±.0112	114.00±56.09
Temp.→HUC	.2810±.0198	**.8216±.0067**	**.0386±.0069**	**.0261±.0022**	68.20±29.15	.5160±.0230	.9570±.0027	**.0389±.0081**	.0247±.0067	21.20±8.32
Vec.→HUC	.2747±.0180	.8149±.0071	.0493±.0109	.0316±.0042	52.40±33.13	.5241±.0156	.9574±.0027	.0400±.0094	**.0245±.0052**	12.00±8.40
Dir.→HUC	.2660±.0290	.7847±.0085	.1068±.0193	.0515±.0114	10.80±19.51	.5079±.0140	.9549±.0022	.0695±.0267	.0455±.0090	0.00±0.00

K.3 SUPPORT2 (KNAUS ET AL., 1995)

Data and label hierarchy. SUPPORT2 contains 9,105 seriously ill hospitalized adults from five centers.[2] We use a stratified subset of $5{,}000$ patients with an observed two-month functional outcome. The five outcome categories define the leaves $y_1, \ldots, y_5$. Each of five data splits contains $2{,}250$ training, $1{,}001$ calibration, 750 validation, and 999 test examples.

The root v_0 separates survival from death within two months. Among survivors, v_1 separates milder from major limitations. Node v_2 separates y_1 and y_2, and v_3 separates y_3 and y_4. Thus $V^\circ = \{v_0, v_1, v_2, v_3\}$.

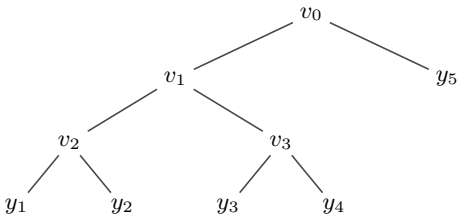


Figure 5: SUPPORT2 outcome tree. The leaves are: y_1, no moderate-or-greater limitation; y_2, assistance with daily activities; y_3, major functional limitation; y_4, intubation or coma; y_5, death within two months. The hierarchy and utilities are synthetic, not clinical recommendations.

Features. There are 30 input variables before categorical expansion: 24 numerical variables and six categorical variables. The categorical variables are sex, disease group, disease class, income, race, and cancer status. The numerical variables describe patient background and baseline physiological and laboratory measurements. Outcome variables, post-outcome variables, costs, and precomputed prognostic scores are excluded. Numerical variables are standardized after median imputation and are accompanied by missing-value indicators; categorical variables are represented by indicators. The resulting indicators and numerical variables form the classifier input; the output is a probability vector over five leaves.

The six base predictors in Appendix K.1 are trained on these features to predict the five leaf labels.

[2]Data obtained from `http://hbiostat.org/data` courtesy of the Vanderbilt University Department of Biostatistics.

Utilities. These utilities use the same binary action-comparison construction as the Toy decision utilities. The interpretation of the outcome sets, thresholds, and action payoffs is given in Appendix J.2. For decisions $j \in \{1,2,3\}$, let D_j be the leaves on which the decision is active and $S_j \subseteq D_j$ the leaves for which action 1 is preferred:

j	D_j	S_j	two actions
1	$\{y_1, y_2, y_3, y_4, y_5\}$	$\{y_5\}$	prepare for death within two months / usual response
2	$\{y_1, y_2\}$	$\{y_2\}$	prepare assistance with daily living / usual follow-up
3	$\{y_3, y_4\}$	$\{y_4\}$	prepare intensive support / usual support

(206)

For $\tau \in \Theta := \{0.25, 0.50, 0.75\}$, define

$$d_j^\tau(y) = \mathbf{1}\{y \in S_j\} - \tau \mathbf{1}\{y \in D_j\}, \qquad U_j^\tau(a,y) = (2a-1)d_j^\tau(y), \quad a \in \{0,1\}, \tag{207}$$

and let

$$\widehat{a}_j^\tau(p) := \operatorname*{arg\,max}_{\prec;\, a \in \{0,1\}} \sum_{y \in \mathcal{Y}} p_y U_j^\tau(a,y), \tag{208}$$

$$u_j^\tau(p,y) := U_j^\tau\left(\widehat{a}_j^\tau(p), y\right). \tag{209}$$

The fixed order $\prec$ resolves ties. These synthetic utilities encode three action comparisons at three parameter values, not treatment effects or clinical guidelines.

For $A \subseteq \mathcal{Y}$, write $p(A) := \sum_{z \in A} p_z$. Averaging the selected action's payoff gives

$$\mu_{u_j^\tau}(p) = \sum_{z \in \mathcal{Y}} p_z u_j^\tau(p,z) = \{2\widehat{a}_j^\tau(p) - 1\}\{p(S_j) - \tau p(D_j)\} = |p(S_j) - \tau p(D_j)|.$$

The last equality holds because $\widehat{a}_j^\tau(p)$ maximizes the predicted expected payoff. Thus the three decisions have predicted means

$$\mu_{u_1^\tau}(p) = |p_{y_5} - \tau|, \qquad \mu_{u_2^\tau}(p) = |p_{y_2} - \tau(p_{y_1} + p_{y_2})|, \qquad \mu_{u_3^\tau}(p) = |p_{y_4} - \tau(p_{y_3} + p_{y_4})|.$$

These are expected payoffs, not probabilities of correct classification. The average is over all five outcomes; for decisions 2 and 3, outcomes outside D_j contribute zero, without division by $p(D_j)$.

The main comparison uses all nine utilities, $\mathcal{U} = \mathcal{U}^{(3)} := \{u_j^\tau : j \in \{1,2,3\},\ \tau \in \Theta\}$, whose relevant internal nodes cover $\{v_0, v_1, v_2, v_3\}$.

The SUPPORT2 results in the main-text Table 1 use $\mathcal{U} = \mathcal{U}^{(3)}$: all three decisions at all three thresholds are included in both correction and UC/HUC evaluation.

Subgroups. Each subgroup is represented by $c_A(x) = \mathbf{1}\{x \in A\}$. There are ten candidate indicators: one for the whole population, three for age (below 65, 65 to below 80, and at least 80 years), two for sex, and four for disease class (ARF/MOSF, COPD/CHF/cirrhosis, coma, and cancer). The indicators are evaluated from the corresponding patient attributes; neither predicted labels nor prediction errors define membership. These groups may overlap, but their intersections are not added. Missing age values belong to none of the three age groups.

The whole-population indicator is always included. Each other indicator is retained when its group contains at least five observations in both calibration and validation. Thus the retained family satisfies $|\mathcal{C}| \leq 10$. The same retained family is used by all methods and utility classes within a split, with no further selection on test data.

K.3.1 Main Results

Table 8 gives the pooled results for $\mathcal{U}^{(3)}$ over the six base predictors and five data splits. This is the nine-utility SUPPORT2 setting used in Table 1; the present table additionally includes HUC-Boost (0.05) as a supplementary step-size comparison. HUC-Boost reduces mean HUC relative to Baseline and gives lower mean HUC than the fixed-step correction. The fixed-step correction retains more updates and has higher mean UC and lower mean Accuracy.

Results by base predictor. Tables 9 and 10 give the full base-predictor results for the nine-utility class used in the main text. The fixed step is especially poor for GNB and DT and uses more updates than HUC-Boost for every base predictor. RF and XGB give small base-specific exceptions in HUC, but these exceptions do not change the pooled conclusion that the fixed step is inferior on SUPPORT2.

Table 8: Complete SUPPORT2 results for the nine-utility class $\mathcal{U}^{(3)}$. Values are mean ± sample standard deviation over six base predictors and five data splits. UC-Boost and HUC-Boost are abbreviated as UC-B and HUC-B, respectively.

Method	Acc. ↑	AUC ↑	UC ↓	HUC ↓	Updates
Baseline	.4990±.1994	.6763±.0828	.1671±.1777	.1481±.1422	0.00±0.00
UC-B	.5746±.0506	.6813±.0789	.0394±.0139	.0504±.0313	33.90±39.40
HUC-B	.5754±.0453	.6985±.0600	.0364±.0101	.0351±.0089	45.20±46.89
HUC-B (0.05)	.5023±.1967	.6844±.0787	.1466±.1828	.1270±.1444	198.80±148.38
Temp.	.4990±.1994	.6865±.0722	.0769±.0692	.0757±.0693	0.00±0.00
Temp.→HUC	.5872±.0319	.7031±.0568	.0324±.0064	.0323±.0062	20.40±22.61
Vec.	.5750±.0532	.6907±.0654	.0380±.0154	.0380±.0154	0.00±0.00
Vec.→HUC	.5886±.0322	.7005±.0505	**.0314±.0081**	**.0309±.0079**	14.27±12.55
Dir.	.5794±.0457	.7125±.0657	.0351±.0129	.0351±.0128	0.00±0.00
Dir.→HUC	**.5889±.0322**	**.7181±.0500**	.0323±.0077	.0322±.0072	13.27±14.08

Table 9: SUPPORT2 ($\mathcal{U}^{(3)}$): baseline methods by base predictor. Values are mean ± sample standard deviation over five data splits. UC-Boost and HUC-Boost are abbreviated as UC-B and HUC-B, respectively.

	LR					**RF**				
Method	Acc. ↑	AUC ↑	UC ↓	HUC ↓	Updates	Acc. ↑	AUC ↑	UC ↓	HUC ↓	Updates
Baseline	.6088±.0125	.7621±.0218	.0385±.0075	.0385±.0075	0.00±0.00	**.6152±.0053**	.7462±.0289	.0473±.0083	.0473±.0083	0.00±0.00
UC-B	.6088±.0120	.7614±.0210	.0255±.0019	.0255±.0019	3.80±2.17	**.6152±.0128**	.7436±.0289	.0350±.0104	.0350±.0104	7.60±10.41
Temp.	.6088±.0125	.7621±.0216	.0284±.0055	.0268±.0067	0.00±0.00	**.6152±.0053**	.7442±.0284	.0407±.0083	.0407±.0083	0.00±0.00
Vec.	.6094±.0119	.7633±.0217	.0269±.0035	.0269±.0035	0.00±0.00	.6140±.0038	.7286±.0394	**.0306±.0071**	**.0306±.0071**	0.00±0.00
Dir.	**.6100±.0120**	**.7677±.0182**	**.0251±.0028**	**.0251±.0028**	0.00±0.00	.6124±.0087	**.7475±.0133**	.0310±.0095	.0310±.0095	0.00±0.00
	GNB					**XGB**				
Method	Acc. ↑	AUC ↑	UC ↓	HUC ↓	Updates	Acc. ↑	AUC ↑	UC ↓	HUC ↓	Updates
Baseline	.0701±.0131	.5729±.0136	.5091±.0272	.3978±.0098	0.00±0.00	.6036±.0066	.7406±.0207	.0993±.0088	.0993±.0088	0.00±0.00
UC-B	**.4987±.0361**	.5828±.0288	.0603±.0051	.1099±.0144	98.20±2.39	.6034±.0054	.7512±.0132	.0354±.0092	.0354±.0092	16.40±7.64
Temp.	.0701±.0131	.6206±.0250	.2101±.0046	.2101±.0046	0.00±0.00	.6036±.0066	.7446±.0198	.0316±.0086	.0311±.0092	0.00±0.00
Vec.	.4821±.0440	**.6289±.0263**	.0605±.0138	.0605±.0138	0.00±0.00	.6086±.0093	.7466±.0154	.0307±.0094	.0307±.0094	0.00±0.00
Dir.	.4985±.0239	.6142±.0239	**.0559±.0149**	**.0559±.0149**	0.00±0.00	**.6118±.0085**	**.7637±.0152**	**.0292±.0065**	**.0297±.0059**	0.00±0.00
	DT					**MLP**				
Method	Acc. ↑	AUC ↑	UC ↓	HUC ↓	Updates	Acc. ↑	AUC ↑	UC ↓	HUC ↓	Updates
Baseline	.5007±.0194	.5820±.0165	.2745±.0275	.2745±.0275	0.00±0.00	.5956±.0174	.6538±.0444	.0337±.0088	.0314±.0075	0.00±0.00
UC-B	.5209±.0197	.5979±.0225	.0501±.0073	.0667±.0080	71.00±27.46	**.6008±.0180**	.6509±.0375	.0299±.0057	.0301±.0053	6.40±10.57
Temp.	.5007±.0194	.5939±.0225	.1203±.0097	.1168±.0128	0.00±0.00	.5956±.0174	.6537±.0443	.0305±.0073	.0286±.0031	0.00±0.00
Vec.	.5407±.0188	.6104±.0326	.0515±.0079	.0515±.0079	0.00±0.00	.5950±.0201	.6664±.0337	**.0276±.0049**	**.0276±.0049**	0.00±0.00
Dir.	**.5469±.0174**	**.6388±.0320**	**.0395±.0044**	**.0395±.0044**	0.00±0.00	.5970±.0155	**.7428±.0183**	.0298±.0059	.0297±.0061	0.00±0.00

Table 10: SUPPORT2 ($\mathcal{U}^{(3)}$): HUC-Boost results by base predictor. Values are mean ± sample standard deviation over five data splits. UC-Boost and HUC-Boost are abbreviated as UC-B and HUC-B, respectively.

	LR					**RF**				
Method	Acc. ↑	AUC ↑	UC ↓	HUC ↓	Updates	Acc. ↑	AUC ↑	UC ↓	HUC ↓	Updates
HUC-B	.6086±.0121	.7580±.0208	.0282±.0019	.0271±.0017	12.80±7.16	.6100±.0166	.7459±.0397	.0342±.0108	.0342±.0107	19.20±12.77
HUC-B (0.05)	**.6112±.0138**	.7608±.0197	.0285±.0034	.0285±.0034	72.80±54.21	.6156±.0109	.7485±.0314	.0332±.0095	.0332±.0095	120.00±90.77
Temp.→HUC	.6098±.0117	.7597±.0200	.0303±.0052	.0303±.0052	9.60±15.13	.6158±.0097	**.7499±.0330**	.0306±.0070	.0318±.0067	16.80±17.30
Vec.→HUC	.6088±.0112	.7555±.0230	.0296±.0062	.0278±.0038	8.00±10.20	**.6162±.0117**	.7281±.0297	**.0305±.0077**	**.0301±.0082**	16.80±12.46
Dir.→HUC	.6098±.0118	**.7641±.0171**	**.0256±.0033**	**.0256±.0033**	0.80±1.79	.6138±.0097	.7267±.0230	.0314±.0071	.0316±.0071	20.00±12.33
	GNB					**XGB**				
Method	Acc. ↑	AUC ↑	UC ↓	HUC ↓	Updates	Acc. ↑	AUC ↑	UC ↓	HUC ↓	Updates
HUC-B	.5075±.0185	.6407±.0165	.0443±.0107	.0421±.0065	86.40±4.56	.6042±.0069	.7484±.0111	.0348±.0077	.0347±.0077	22.40±8.76
HUC-B (0.05)	.0809±.0247	.5756±.0171	.5067±.0277	.3901±.0190	367.20±73.34	.6058±.0110	.7490±.0109	.0318±.0062	.0318±.0062	141.60±48.71
Temp.→HUC	.5483±.0198	.6441±.0409	**.0317±.0033**	**.0307±.0032**	16.00±7.48	.6048±.0133	.7458±.0124	.0361±.0088	.0361±.0088	19.20±27.77
Vec.→HUC	.5451±.0236	.6556±.0352	.0376±.0137	.0376±.0137	18.40±7.80	.6098±.0075	.7430±.0124	**.0286±.0038**	**.0286±.0038**	9.60±8.29
Dir.→HUC	**.5485±.0218**	**.6655±.0234**	.0362±.0092	.0364±.0091	23.20±13.68	**.6112±.0082**	**.7629±.0141**	.0295±.0070	.0299±.0064	6.40±6.69
	DT					**MLP**				
Method	Acc. ↑	AUC ↑	UC ↓	HUC ↓	Updates	Acc. ↑	AUC ↑	UC ↓	HUC ↓	Updates
HUC-B	.5241±.0160	.6256±.0227	.0465±.0076	.0443±.0052	124.80±27.33	.5980±.0143	.6721±.0333	**.0302±.0063**	**.0282±.0037**	5.60±8.29
HUC-B (0.05)	.4981±.0220	.6089±.0274	.2463±.0260	.2463±.0260	398.40±2.19	**.6024±.0104**	.6637±.0471	.0333±.0093	.0320±.0077	92.80±98.48
Temp.→HUC	.5481±.0148	.6454±.0208	.0335±.0052	.0335±.0052	41.60±29.61	.5962±.0203	.6738±.0317	.0325±.0088	.0317±.0078	19.20±26.89
Vec.→HUC	**.5520±.0151**	**.6582±.0267**	**.0312±.0047**	**.0316±.0053**	28.80±14.81	.5998±.0145	.6626±.0381	.0310±.0093	.0301±.0082	4.00±5.66
Dir.→HUC	.5489±.0180	.6503±.0300	.0371±.0072	.0369±.0074	19.60±21.00	.6010±.0162	**.7394±.0253**	.0341±.0081	.0329±.0049	9.60±10.43

K.3.2 Additional Results

The nested classes add one decision at a time, including all three thresholds for that decision:

$$\mathcal{U}^{(1)} = \{u_1^{0.25}, u_1^{0.50}, u_1^{0.75}\},$$
$$\mathcal{U}^{(2)} = \mathcal{U}^{(1)} \cup \{u_2^{0.25}, u_2^{0.50}, u_2^{0.75}\},$$
$$\mathcal{U}^{(3)} = \mathcal{U}^{(2)} \cup \{u_3^{0.25}, u_3^{0.50}, u_3^{0.75}\}.$$

Thus $\mathcal{U}^{(1)}$ contains only the comparison concerning preparation for death within two months; $\mathcal{U}^{(2)}$ adds the comparison concerning assistance with daily living; and $\mathcal{U}^{(3)}$ additionally includes the intensive-support comparison. Their numbers of utilities are $3, 6, 9$. The unions of their relevant internal nodes are, respectively, $\{v_0\}$, $\{v_0, v_1, v_2\}$, and $\{v_0, v_1, v_2, v_3\}$, of sizes $1, 3, 4$.

Utility-class expansion. Table 11 compares $\mathcal{U}^{(1)}$, $\mathcal{U}^{(2)}$, and $\mathcal{U}^{(3)}$, varying the target class rather than fixing it to $\mathcal{U}^{(3)}$ as in the main-text comparison. HUC-Boost keeps the resulting HUC at a similar level as the class expands, while the number of selected updates grows with the number of target utilities and relevant nodes. For the two larger classes, HUC-Boost yields lower mean HUC than direct UC-Boost. The fixed-step correction has higher mean HUC for every pooled class, and its retained update count grows as the class expands.

Table 11: Supplementary SUPPORT2 utility-class expansion. Each correction is fitted separately from Baseline for each target class. HUC-B uses $\alpha = \Gamma/\Lambda$; HUC-B (0.05) uses constant step magnitude 0.05. UC-Boost and HUC-Boost are abbreviated as UC-B and HUC-B, respectively.

Target class	Utilities	Relevant nodes	HUC error after correction			Updates		
			UC-B	HUC-B	HUC-B (0.05)	UC-B	HUC-B	HUC-B (0.05)
$\mathcal{U}^{(1)}$	3	1	.0333±.0083	.0343±.0081	.1264±.1451	11.57±11.41	11.90±11.79	47.77±36.81
$\mathcal{U}^{(2)}$	6	3	.0557±.0431	.0346±.0081	.1268±.1448	29.30±34.74	32.70±35.19	138.60±108.89
$\mathcal{U}^{(3)}$	9	4	.0504±.0313	.0351±.0089	.1270±.1444	33.90±39.40	45.20±46.89	198.80±148.38

K.4 MASSIVE (FitzGerald et al., 2023; Bastianelli et al., 2020)

Data, features, and label hierarchy. MASSIVE contains $16{,}521$ English utterances with 60 intent labels grouped into 18 scenarios. The root has one child for each scenario, and the children of each scenario are its intent labels. Thus every root-to-leaf path has the form root–scenario–intent. If $\mathcal{S}$ denotes the set of 18 scenario nodes, then

$$\mathrm{ch}(\rho) = \mathcal{S}, \qquad \mathrm{ch}(s) = \mathcal{Y}_s \quad (s \in \mathcal{S}), \qquad \mathcal{Y} = \bigsqcup_{s \in \mathcal{S}} \mathcal{Y}_s, \quad |\mathcal{Y}| = 60. \tag{210}$$

Each utterance is represented by TF–IDF (term frequency–inverse document frequency) features. TF–IDF weights term frequencies within each utterance by inverse document frequency, reducing the influence of terms shared by many utterances (Manning et al., 2008). We concatenate word unigram and bigram TF–IDF features with character three- to five-gram TF–IDF features. The two vocabularies are restricted to at most $20{,}000$ and $30{,}000$ features, respectively, and logistic regression is trained as the base predictor. We compare post-hoc calibration methods using the probability outputs of this TF–IDF logistic-regression predictor as their common input.

Data allocation. The official training data are divided in proportions $0.60/0.20/0.20$ for base training, initial recalibration, and correction by UC-Boost or HUC-Boost. The development data are divided equally for initial-recalibrator selection and validation; the official test data are used for evaluation. Results are averaged over five data splits.

Utilities. For prediction p, first choose the most probable scenario and then the most probable intent inside it:

$$\widehat{s}(p) = \operatorname*{arg\,max}_{\prec;\, s \in \mathcal{S}} \pi_s(p), \qquad \widehat{y}(p) = \operatorname*{arg\,max}_{\prec;\, y \in \mathcal{Y}_{\widehat{s}(p)}} p_y. \tag{211}$$

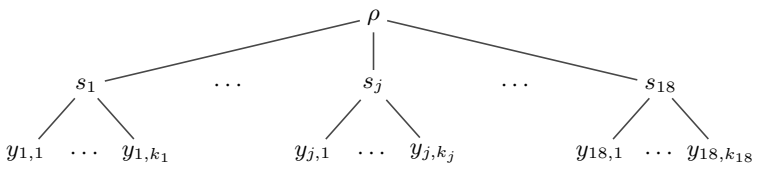


Figure 6: MASSIVE label tree. Dots omit scenario and intent nodes. The intents below s_j are $\mathcal{Y}_{s_j} = \{y_{j,1}, \ldots, y_{j,k_j}\}$, with $\sum_{j=1}^{18} k_j = 60$; each intent belongs to exactly one scenario.

The scenario confidence is $\pi_{\widehat{s}(p)}(p)$, and the intent confidence is the conditional probability $p_{\widehat{y}(p)}/\pi_{\widehat{s}(p)}(p)$, not its unconditional leaf probability. For each pair $(\tau_s, \tau_i) \in \{0.55, 0.65, 0.75\}^2$, the action is

$$a_{\tau_s,\tau_i}(p) = \begin{cases} \text{abstain}, & \pi_{\widehat{s}(p)}(p) < \tau_s, \\ \widehat{s}(p), & \pi_{\widehat{s}(p)}(p) \geq \tau_s \text{ and } p_{\widehat{y}(p)}/\pi_{\widehat{s}(p)}(p) < \tau_i, \\ \widehat{y}(p), & \pi_{\widehat{s}(p)}(p) \geq \tau_s \text{ and } p_{\widehat{y}(p)}/\pi_{\widehat{s}(p)}(p) \geq \tau_i. \end{cases} \tag{212}$$

For a realized intent y, the resulting utility is

$$u_{\tau_s,\tau_i}(p, y) = \begin{cases} 0, & a_{\tau_s,\tau_i}(p) = \text{abstain}, \\ \frac{1}{4}, & a_{\tau_s,\tau_i}(p) = \widehat{s}(p),\ y \in \mathcal{Y}_{\widehat{s}(p)}, \\ -1, & a_{\tau_s,\tau_i}(p) = \widehat{s}(p),\ y \notin \mathcal{Y}_{\widehat{s}(p)}, \\ 1, & a_{\tau_s,\tau_i}(p) = \widehat{y}(p),\ y = \widehat{y}(p), \\ -\frac{1}{2}, & a_{\tau_s,\tau_i}(p) = \widehat{y}(p),\ y \in \mathcal{Y}_{\widehat{s}(p)} \setminus \{\widehat{y}(p)\}, \\ -1, & a_{\tau_s,\tau_i}(p) = \widehat{y}(p),\ y \notin \mathcal{Y}_{\widehat{s}(p)}. \end{cases} \tag{213}$$

Hence $|\mathcal{U}| = 9$. The fixed alphabetical order $\prec$ on scenario and intent labels resolves all ties.

Write $s_p := \pi_{\widehat{s}(p)}(p)$ and $t_p := p_{\widehat{y}(p)}$. The predicted masses of the correct intent, the other intents in the selected scenario, and the intents outside that scenario are t_p, $s_p - t_p$, and $1 - s_p$, respectively. Averaging Eq. (213) therefore gives

$$\begin{aligned} \mu_{u_{\tau_s,\tau_i}}(p) &= \sum_{z \in \mathcal{Y}} p_z u_{\tau_s,\tau_i}(p, z) \\ &= \begin{cases} 0, & a_{\tau_s,\tau_i}(p) = \text{abstain}, \\ \frac{1}{4} s_p - (1 - s_p), & a_{\tau_s,\tau_i}(p) = \widehat{s}(p), \\ t_p - \frac{1}{2}(s_p - t_p) - (1 - s_p), & a_{\tau_s,\tau_i}(p) = \widehat{y}(p). \end{cases} \end{aligned}$$

The scenario and intent expressions simplify to $\frac{5}{4} s_p - 1$ and $\frac{3}{2} t_p + \frac{1}{2} s_p - 1$, respectively. This is the predicted mean payoff of the threshold-based reporting rule, rather than a correctness probability or the conditional intent confidence t_p/s_p.

Subgroups. The family has $|\mathcal{C}| = 3$: $c \equiv 1$, $\mathbf{1}\{\text{word count} \leq m\}$, and $\mathbf{1}\{\text{word count} > m\}$, where m is the training median of word count.

Results. Table 12 reports all ten methods. Both HUC-Boost and the fixed-step correction reduce mean HUC relative to Baseline. This dataset does not show the same ordering between the two update rules as the main experiments.

K.5 NSL-KDD (TAVALLAEE ET AL., 2009)

Data, features, and label hierarchy. NSL-KDD[3] uses 25,192 records with five leaf labels: normal, dos, probe, r2l, and u2r. The root separates normal traffic from attacks, and the attack node separates the four attack families. The 41 input attributes give 118 numerical features after categorical encoding. The six base predictors use five stratified data splits, with training/calibration/validation/test proportions $0.45/0.20/0.15/0.20$.

[3]NSL-KDD dataset: `https://www.unb.ca/cic/datasets/nsl.html`.

Table 12: MASSIVE results. Values are mean ± sample standard deviation over five data splits of the TF–IDF logistic-regression predictor. HUC-B (0.05) uses constant step magnitude 0.05. UC-Boost and HUC-Boost are abbreviated as UC-B and HUC-B, respectively.

Method	Acc. ↑	AUC ↑	UC ↓	HUC ↓	Updates
Baseline	.8222±.0025	.9913±.0008	.1380±.0039	.1190±.0035	0.00±0.00
UC-B	.8223±.0026	.9899±.0009	.0216±.0077	.0142±.0043	44.60±33.25
HUC-B	.8197±.0036	.9891±.0009	.0302±.0078	.0170±.0065	251.60±50.97
HUC-B (0.05)	.8223±.0027	.9897±.0009	**.0213±.0083**	**.0140±.0071**	395.80±66.34
Temp.	.8222±.0025	.9906±.0007	.0335±.0075	.0189±.0040	0.00±0.00
Temp.→HUC	.8206±.0029	.9904±.0008	.0371±.0057	.0181±.0048	109.40±110.03
Vec.	**.8295±.0010**	.9909±.0009	.0373±.0066	.0178±.0016	0.00±0.00
Vec.→HUC	.8268±.0014	.9904±.0009	.0476±.0087	.0235±.0070	183.00±71.87
Dir.	.8286±.0038	**.9917±.0008**	.0714±.0042	.0505±.0035	0.00±0.00
Dir.→HUC	.8260±.0025	.9908±.0007	.0338±.0128	.0229±.0075	224.80±109.76

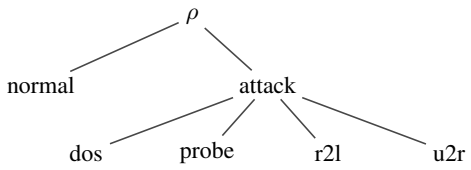


Figure 7: NSL-KDD label tree. The five leaves are normal traffic and the four attack families.

Utilities. For a node v, let $\mathcal{Y}_v$ be its descendant-leaf set and write $J_v(y) := \mathbf{1}\{y \in \mathcal{Y}_v\}$ and $\pi_v(p) := \sum_{z\in\mathcal{Y}_v} p_z$. Thus $J_v(y)$ records whether the returned category contains the true label. Set $p_{\text{normal}} := \pi_{\text{normal}}(p)$. Also set $\pi_{\text{attack}}(p) := p_{\text{dos}} + p_{\text{probe}} + p_{\text{r2l}} + p_{\text{u2r}}$. The root decision is

$$\widehat{v}_0(p) = \begin{cases} \text{normal}, & p_{\text{normal}} \geq \pi_{\text{attack}}(p), \\ \text{attack}, & p_{\text{normal}} < \pi_{\text{attack}}(p). \end{cases}$$

The family decision returns normal when the root selects normal; otherwise it selects the most probable attack leaf:

$$\widehat{v}_1(p) = \begin{cases} \text{normal}, & \widehat{v}_0(p) = \text{normal}, \\ \arg\max_{\prec;\, v\in\{\text{dos,probe,r2l,u2r}\}} p_v, & \widehat{v}_0(p) = \text{attack}. \end{cases}$$

The third decision is $\widehat{y}(p) = \arg\max_{\prec;\, y\in\mathcal{Y}} p_y$ over all five leaves. The three utilities are

$$\begin{aligned} u_{\text{status}}(p, y) &= J_{\widehat{v}_0(p)}(y) = \mathbf{1}\{y \in \mathcal{Y}_{\widehat{v}_0(p)}\}, \\ u_{\text{family}}(p, y) &= \mathbf{1}\{y = \widehat{v}_1(p)\}, \qquad u_{\text{leaf}}(p, y) = \mathbf{1}\{y = \widehat{y}(p)\}. \end{aligned} \tag{214}$$

The corresponding predicted mean utilities are

$$\mu_{u_{\text{status}}}(p) = \sum_{z\in\mathcal{Y}_{\widehat{v}_0(p)}} p_z = \pi_{\widehat{v}_0(p)}(p), \qquad \mu_{u_{\text{family}}}(p) = \sum_{z\in\mathcal{Y}} p_z \mathbf{1}\{z = \widehat{v}_1(p)\} = p_{\widehat{v}_1(p)},$$
$$\mu_{u_{\text{leaf}}}(p) = \sum_{z\in\mathcal{Y}} p_z \mathbf{1}\{z = \widehat{y}(p)\} = p_{\widehat{y}(p)}.$$

The status mean is the total probability of the selected normal/attack category. Since $\widehat{v}_1(p)$ is a leaf in this tree, the family mean is its unconditional probability, not its probability conditional on attack. The leaf mean is the probability of the globally most probable label.

Thus $|\mathcal{U}| = 3$. The family and global-leaf decisions need not coincide: the root can select attack while the single most probable leaf is normal. Ties at the root select normal; other ties use the order normal, dos, probe, r2l, u2r.

Subgroups. The nine indicators ($|\mathcal{C}| = 9$) are the whole population, the three protocol groups, the three most frequent training services, and the two groups with source-byte count at most and greater than its training median. Protocol groups correspond to tcp, udp, and icmp. Each indicator tests only its stated attribute condition; no intersections are added.

Table 13: NSL-KDD results. Values are mean ± sample standard deviation over six base predictors and five data splits. HUC-B (0.05) uses constant step magnitude 0.05. UC-Boost and HUC-Boost are abbreviated as UC-B and HUC-B, respectively.

Method	Acc. ↑	AUC ↑	UC ↓	HUC ↓	Updates
Baseline	.9504±.1004	.9577±.0653	.0471±.1011	.0452±.0980	0.00±0.00
UC-B	.9816±.0244	.9687±.0444	.0039±.0034	.0035±.0031	62.07±39.19
HUC-B	.9801±.0283	.9672±.0514	.0039±.0031	.0033±.0026	92.37±69.38
HUC-B (0.05)	.9505±.1004	.9621±.0605	.0448±.1014	.0432±.0983	85.37±75.29
Temp.	.9504±.1004	.9619±.0563	.0285±.0555	.0275±.0552	0.00±0.00
Temp.→HUC	.9838±.0195	.9766±.0363	.0036±.0028	.0032±.0026	64.43±66.43
Vec.	.9735±.0429	.9647±.0553	.0129±.0195	.0125±.0196	0.00±0.00
Vec.→HUC	**.9844±.0181**	**.9812±.0324**	**.0033±.0025**	**.0029±.0023**	58.93±58.73
Dir.	.9738±.0411	.9652±.0465	.0115±.0178	.0112±.0177	0.00±0.00
Dir.→HUC	.9842±.0176	.9690±.0400	**.0033±.0023**	**.0029±.0020**	44.23±55.22

Results. Table 13 reports all ten methods, with means and sample standard deviations over the six base predictors and five splits.

K.6 N-BaIoT (Meidan et al., 2018)

Data, features, and label hierarchy. The N-BaIoT[4] experiment uses 1,000 records, approximately balanced across 11 leaf labels: benign traffic and five attack types for each of Gafgyt and Mirai. The root separates benign traffic from attacks; the attack node separates the two families, whose children are their respective attack types. Classifiers use the first ten numerical features. The six base predictors and training/calibration/validation/test proportions match those for NSL-KDD.

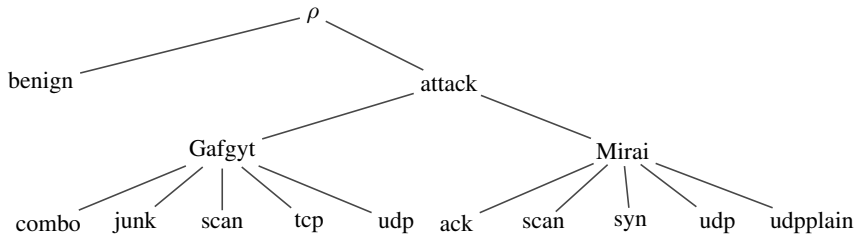


Figure 8: N-BaIoT label tree. Identically named attack types under different families are different leaves; for example, Gafgyt scan and Mirai scan are distinct labels.

Utilities. For a node v, let $\mathcal{Y}_v$ be its descendant-leaf set and write $J_v(y) := \mathbf{1}\{y \in \mathcal{Y}_v\}$ and $\pi_v(p) := \sum_{z\in\mathcal{Y}_v} p_z$. The indicator is one exactly when the true label lies in the returned node's subtree. The root decision $\widehat{v}_0(p)$ is benign if $p_{\text{benign}} \geq \pi_{\text{attack}}(p)$ and attack otherwise. Conditional on choosing attack, the family decision compares the total masses of the Gafgyt and Mirai subtrees:

$$\widehat{v}_1(p) = \begin{cases} \text{benign}, & \widehat{v}_0(p) = \text{benign}, \\ \arg\max_{\prec;\, v\in\{\text{Gafgyt,Mirai}\}} \pi_v(p), & \widehat{v}_0(p) = \text{attack}. \end{cases}$$

The third decision $\widehat{y}(p) = \arg\max_{\prec;\, y\in\mathcal{Y}} p_y$ selects the most probable of all 11 leaves, independently of these two preceding decisions. The utilities are

$$\begin{aligned} u_{\text{status}}(p,y) &= J_{\widehat{v}_0(p)}(y) = \mathbf{1}\{y \in \mathcal{Y}_{\widehat{v}_0(p)}\}, \\ u_{\text{family}}(p,y) &= J_{\widehat{v}_1(p)}(y) = \mathbf{1}\{y \in \mathcal{Y}_{\widehat{v}_1(p)}\}, \\ u_{\text{type}}(p,y) &= \mathbf{1}\{y = \widehat{y}(p)\}. \end{aligned} \tag{215}$$

The corresponding predicted mean utilities are

$$\mu_{u_{\text{status}}}(p) = \sum_{z\in\mathcal{Y}_{\widehat{v}_0(p)}} p_z = \pi_{\widehat{v}_0(p)}(p), \qquad \mu_{u_{\text{family}}}(p) = \sum_{z\in\mathcal{Y}_{\widehat{v}_1(p)}} p_z = \pi_{\widehat{v}_1(p)}(p),$$

$$\mu_{u_{\text{type}}}(p) = \sum_{z\in\mathcal{Y}} p_z \mathbf{1}\{z = \widehat{y}(p)\} = p_{\widehat{y}(p)}.$$

[4]Dataset: UCI Machine Learning Repository, `https://doi.org/10.24432/C5RC8J`.

When an attack family is returned, its mean utility is the total unconditional probability of all attack types in that family, not a probability normalized by $\pi_{\text{attack}}(p)$. When benign is returned, this mean reduces to p_{benign}. The type mean is the probability of the selected leaf among all 11 labels.

Thus $|\mathcal{U}| = 3$. A root tie selects benign and a family tie selects Gafgyt. Global-leaf ties use benign first, then Gafgyt types in the order combo, junk, scan, tcp, udp, and then Mirai types in the order ack, scan, syn, udp, udpplain.

Subgroups. The indicators are the whole population, each device present in training, and the two groups with first-feature value at most and greater than its training median. Thus the family contains one device indicator per represented device and three additional indicators. Device identity is used only for subgroup membership, not as an eleventh input feature. Indicators with identical values, such as the whole-population and device indicators in a single-device sample, are not merged.

Results. Table 14 reports all ten methods, with means and sample standard deviations over the six base predictors and five splits. Both the calibration and validation subsets are small.

Table 14: N-BaIoT results. Values are mean ± sample standard deviation over six base predictors and five data splits. HUC-B (0.05) uses constant step magnitude 0.05. UC-Boost and HUC-Boost are abbreviated as UC-B and HUC-B, respectively.

Method	Acc. ↑	AUC ↑	UC ↓	HUC ↓	Updates
Baseline	.7852±.1560	.9703±.0382	.1021±.0998	.0672±.0633	0.00±0.00
UC-B	.7960±.1262	.9720±.0300	.0338±.0221	.0324±.0197	31.50±30.98
HUC-B	.7987±.1254	.9736±.0307	.0357±.0212	.0278±.0186	35.27±28.58
HUC-B (0.05)	.7960±.1281	.9749±.0283	.0415±.0332	.0325±.0316	137.33±80.47
Temp.	.7852±.1560	.9711±.0396	.0415±.0354	.0391±.0364	0.00±0.00
Temp.→HUC	.8010±.1264	.9764±.0265	.0326±.0239	.0265±.0189	24.80±33.13
Vec.	.8077±.1243	.9770±.0229	.0361±.0293	.0290±.0242	0.00±0.00
Vec.→HUC	.8104±.1133	.9787±.0171	.0363±.0272	.0271±.0160	13.57±22.93
Dir.	**.8380±.0439**	**.9843±.0043**	.0413±.0412	.0269±.0200	0.00±0.00
Dir.→HUC	.8357±.0418	.9841±.0044	**.0311±.0116**	**.0243±.0104**	16.77±23.76