Option 3: Risk-aware balanced recursive partition into axis-aligned rectangles (2D)

Goal. Given calibration data

$$\mathcal{D} = \{(u_{1,k}, u_{2,k}, E_k)\}_{k=1}^{n}, \qquad (u_{1,k}, u_{2,k}) \in [0,1]^2,\ E_k \in \{0,1\},$$

construct B axis-aligned rectangles (leaf regions) with *near-uniform mass* (roughly n/B points per leaf), where splits are chosen to *separate error rates* between children.

Node representation. A node (region) ν stores:

- an index set $S_\nu \subseteq [n]$ of points currently in the node,
- a bounding box $R_\nu = [\ell_{1,\nu}, r_{1,\nu}) \times [\ell_{2,\nu}, r_{2,\nu}) \subseteq [0,1]^2$,
- the empirical error rate $\widehat{\eta}(\nu)\frac{1}{|S_\nu|}\sum_{k\in S_\nu} E_k$.

The root node is ν_0 with $S_{\nu_0} = [n]$ and $R_{\nu_0} = [0,1) \times [0,1)$.

Size constraint (near-uniform mass). Fix

$$n_{\min} \left\lfloor \frac{n}{B} \right\rfloor \quad \text{(e.g., } n = 4000,\ B = 30 \Rightarrow n_{\min} \approx 133).$$

Every split must produce children with at least $n_{\min}$ points.

Candidate thresholds. Fix a small set of quantile levels

$$\mathcal{Q} = \{q_1, \ldots, q_M\} \subset (0,1) \quad \text{(e.g., } \{0.1, 0.2, \ldots, 0.9\}).$$

For a node ν and dimension $d \in \{1,2\}$, define candidate thresholds

$$\tau \in \Big\{\text{Quantile}_q(\{u_{d,k} : k \in S_\nu\}) : q \in \mathcal{Q}\Big\},$$

optionally discarding duplicates (ties).

Split and score. For node ν, a candidate split (d,τ) induces children:

$$S_L = \{k \in S_\nu : u_{d,k} \le \tau\}, \qquad S_R = S_\nu \setminus S_L,$$

with corresponding rectangles (updating only the split coordinate bounds):

$$R_L = R_\nu \cap \{u_d \le \tau\}, \qquad R_R = R_\nu \cap \{u_d > \tau\}.$$

The split is *admissible* if $|S_L| \ge n_{\min}$ and $|S_R| \ge n_{\min}$.

Define empirical child error rates

$$\widehat{\eta}_L = \frac{1}{|S_L|}\sum_{k\in S_L} E_k, \qquad \widehat{\eta}_R = \frac{1}{|S_R|}\sum_{k\in S_R} E_k.$$

A simple, effective split score is the absolute separation:

$$\Delta(d,\tau;\nu)|\widehat{\eta}_L - \widehat{\eta}_R|.$$

Optionally (to discourage very unbalanced splits even when admissible), use

$$\Delta'(d,\tau;\nu)\Delta(d,\tau;\nu) \cdot \sqrt{\frac{|S_L||S_R|}{|S_\nu|^2}}.$$

(Use either Δ or Δ' consistently.)

Algorithm 1 Risk-aware balanced rectangle partition (2D)

Require: Data $\mathcal{D} = \{(u_{1,k}, u_{2,k}, E_k)\}_{k=1}^{n}$, target leaves B, quantiles $\mathcal{Q}$, minimum leaf size $n_{\min}$.
Ensure: A binary tree; leaves define rectangles $\{R_\nu\}_{\nu \in \mathcal{L}}$, with associated S_ν and $\widehat{\eta}(\nu)$.

Initialize root leaf ν_0 with $S_{\nu_0} = [n]$, $R_{\nu_0} = [0,1) \times [0,1)$.
$\mathcal{L} \leftarrow \{\nu_0\}$.
while $|\mathcal{L}| < B$ **do**
  $\mathit{bestLeaf} \leftarrow$ None; $\mathit{bestSplit} \leftarrow$ None; $\mathit{bestScore} \leftarrow -\infty$.
  for all leaves $\nu \in \mathcal{L}$ **do**
    Compute the best admissible split $(d^\star, \tau^\star)$ for ν:
    $\mathit{score}^\star(\nu) \leftarrow \max\{\Delta(d, \tau; \nu) : d \in \{1,2\},\ \tau \in \mathcal{T}_{d,\nu},\ \text{admissible}\}$
    where $\mathcal{T}_{d,\nu}$ are candidate thresholds from quantiles in $\mathcal{Q}$.
    if no admissible split exists for ν **then**
      continue
    end if
    if $\mathit{score}^\star(\nu) > \mathit{bestScore}$ **then**
      $\mathit{bestScore} \leftarrow \mathit{score}^\star(\nu)$
      $\mathit{bestLeaf} \leftarrow \nu$
      $\mathit{bestSplit} \leftarrow (d^\star, \tau^\star)$
    end if
  end for
  if $\mathit{bestLeaf}$ is None **then** ▷ no admissible splits anywhere
    break
  end if
  Split $\mathit{bestLeaf}$ using $\mathit{bestSplit}$ into children ν_L, ν_R:
  $S_{\nu_L} = \{k \in S_{\mathit{bestLeaf}} : u_{d^\star,k} \le \tau^\star\}$, $S_{\nu_R} = S_{\mathit{bestLeaf}} \setminus S_{\nu_L}$
  $R_{\nu_L} = R_{\mathit{bestLeaf}} \cap \{u_{d^\star} \le \tau^\star\}$, $R_{\nu_R} = R_{\mathit{bestLeaf}} \cap \{u_{d^\star} > \tau^\star\}$
  Update leaf set: $\mathcal{L} \leftarrow (\mathcal{L} \setminus \{\mathit{bestLeaf}\}) \cup \{\nu_L, \nu_R\}$.
end while

Tree growth strategy (exactly B leaves). Maintain a set of current leaves $\mathcal{L}$ (initially $\{\nu_0\}$). Repeatedly split *one* current leaf at a time until $|\mathcal{L}| = B$. At each iteration, among all leaves, select the leaf whose *best admissible split* has the largest score.

Output. The final leaves $\mathcal{L}$ define $|\mathcal{L}| \le B$ rectangles $\{R_\nu\}$. For each leaf, store:

$$R_\nu = [\ell_{1,\nu}, r_{1,\nu}) \times [\ell_{2,\nu}, r_{2,\nu}), \qquad \widehat{\eta}(\nu) = \frac{1}{|S_\nu|} \sum_{k \in S_\nu} E_k, \qquad |S_\nu|.$$

In downstream calibration/certification, treat $Q(u_1, u_2) = \nu$ if $(u_1, u_2) \in R_\nu$.

Inference-time assignment. Given a new point (u_1, u_2), traverse the tree from the root: at each internal node with split (d, τ) go left if $u_d \le \tau$ and right otherwise, until a leaf ν is reached. Return $Q(u_1, u_2) = \nu$.

Implementation notes (important for correctness).

- **Ties at thresholds.** Use the convention "left if $u_d \le \tau$" consistently in both training and inference.
- **Quantile computation.** Quantiles are empirical over $\{u_{d,k} : k \in S_\nu\}$; remove duplicates to avoid empty children.
- **Exact B leaves.** The loop increases leaf count by $+1$ each split. If no admissible split exists before reaching B, you stop early.
- **Uniformity.** This enforces a *lower bound* $n_{\min}$; if you want tighter near-uniformity, also enforce an upper bound $|S_{\text{child}}| \le n_{\max}$ (e.g. $n_{\max} \approx \lceil 1.3\, n/B \rceil$) by discarding splits that create too-large leaves.