

Compiling the Plurality Winner of a Random Forest into OBDDs via Margin Absorption

Daniel Amadori¹, Emanuele Chini^{2,1}, Pietro Sala^{1,*} and Emma Maria Sole Tosato³

¹University of Verona, Verona, Italy

²Sapienza University of Rome, Rome, Italy

³Padova Neuroscience Center, University of Padova, Padova, Italy

Abstract

A random forest classifies a tabular input by aggregating the votes of its decision trees, and a natural prerequisite for formal, region-based explanations is a compact symbolic representation of the regions of input space on which each label is the prediction of the forest. We present a relational encoding that compiles, one tree at a time, the regions on which a label is the *plurality* (argmax) winner of the forest into Ordered Binary Decision Diagrams (OBDDs). The encoding keeps, for a fixed pivot label, one signed margin counter per rival, and folds the trees through a transition relation over two copies of the counter state. Its distinguishing device is *margin absorption*: two latch flags, certain-win and certain-lose, fire the moment the contest against the relevant rivals is decided regardless of the trees still to be folded, after which the decided states collapse to a single absorbing state, reducing the intermediate diagram; a complementary drop discipline frees a counter that can no longer be the binding one. A prioritized one-vs-rest peeling then assembles the per-label win and tie regions and yields a sound and complete test for verifying a box-shaped reason. We prove the two latches fire at exactly the right moment, as soon as no outcome of the trees still to be folded can change the contest and not a step before, since a lead equal to the number of remaining trees can still be driven to a tie; we establish a collapse invariant, showing that once a state is decided it no longer depends on its counters; and we show the assembled win regions coincide with the prediction the forest itself makes. On tuned forests we confirm the construction (the regions are bit-identical with the collapse on and off) and find the reduction grows with the diagram, to several-fold on the densest forests. The encoding is purely relational, in the symbolic model-checking style, and keeps no auxiliary state beyond the counters and flags, the lone auxiliary of the drop phase being projected away within each tree step, and is the compilation core of a compile-once, read-many explainer.

Keywords

Random forests, Formal explanations, OBDD, Symbolic transition relations

1. Introduction

Random forests [1] are among the most widely deployed classifiers on tabular data, combining many decision trees by a vote to achieve strong accuracy. The same ensembling makes a prediction opaque: it is the aggregate of hundreds of threshold tests scattered across the trees, and no single tree, feature, or path accounts for it. As these models are used in settings where a decision must be justified, such as clinical diagnosis [2] and credit scoring [3], the question of *why* a forest assigns a label has become central to trustworthy machine learning.

Heuristic feature-attribution methods such as SHAP [4], LIME [5], and Anchors [6] answer this question approximately, with no guarantee of faithfulness. Formal, abductive explainability instead seeks provably correct explanations: an abductive explanation is a subset-minimal set of feature values that guarantees the prediction [7, 8, 9], and the reasons-behind-decisions framework characterizes sufficient reasons through knowledge compilation [10, 11]. A recurring idea in the compilation-based branch is to pay the cost of formal reasoning once, by compiling a representation of the model from which many queries are then read cheaply, rather than solving a fresh optimization for each sample.

ICTCS 2026: 26th Italian Conference on Theoretical Computer Science, 2026, Udine, Italy

*Corresponding author.

✉ daniel.amadori@univr.it (D. Amadori); emanuele.chini@uniroma1.it (E. Chini); pietro.sala@univr.it (P. Sala); emmamariasole.tosato@studenti.unipd.it (E. M. S. Tosato)

🌐 <https://pnc.unipd.it/phd-students/tosato-emma-maria-sole/> (E. M. S. Tosato)

🆔 0009-0003-4373-7811 (D. Amadori); 0009-0002-9125-4171 (E. Chini); 0000-0002-2612-1519 (P. Sala)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

The prerequisite for such a representation, in the region-based setting we target, is an Ordered Binary Decision Diagram (OBDD) [12] for each label, whose models are exactly the discretized inputs the forest assigns that label. Building these diagrams is the bottleneck: a forest does not decide a label until its trees are tallied, and a symbolic tally must, at every intermediate step, represent every partial vote count consistent with the inputs seen so far, so the intermediate diagram tends to blow up well before all trees are folded. The contribution of this paper is an encoding whose state and transition relation are designed to keep that intermediate diagram small.

The construction is a chain of exact reformulations: no step approximates, each discards information the verdict cannot depend on, and decision diagrams enter only at the last step. The first step, the *cell discretization*, is preliminary and standard: a forest reads each feature only through finitely many thresholds, so inputs that fall on the same side of every threshold are indistinguishable to the forest, and quotienting the input space by this indistinguishability turns a geometric problem over the continuum into a finite combinatorial one. From then on, threshold magnitudes and numerical precision play no further role. Three design choices then shape the state of the encoding. First, we compile the *plurality* (argmax) winner of the forest, the label with strictly more votes than every other, with ties broken by a fixed priority order, rather than a strict-majority verdict; the plurality winner is the actual prediction of the forest whenever a unique top label exists. Second, instead of one counter per label we keep, for a fixed pivot label, one *signed margin counter* per rival, the difference between the votes of the rival and those of the pivot; the contest is then decided by the signs of these margins. Third, and centrally, we equip the state with two *absorption* flags, certain-win and certain-lose, that latch as soon as the margins guarantee the outcome regardless of the trees still to be folded. Once a state is decided, its counters become free and all decided states collapse to a single absorbing state per flag, which reduces the intermediate diagram (Section 6); a drop discipline applies the same idea to individual rivals. The per-label regions are recovered from the pivot accumulators by a prioritized one-vs-rest peeling, which also yields a direct test for verifying a box-shaped reason.

The encoding is purely relational, in the symbolic model-checking style [13, 14]: each tree is a transition relation over two copies of the counter state, and folding the forest is a sequence of relational products. Beyond the counters and flags the encoding keeps no persistent state, the single auxiliary of the drop phase being projected away within the same tree step, and every relation used, comparators, increments, copies, and threshold tests, compiles to a Boolean formula over the counter bits of size polynomial in their number. We develop the encoding and prove its correctness: the exact absorption thresholds, an invariant that a decided state does not depend on its counters, and that the assembled win regions equal the argmax of the forest. We then validate the construction empirically and measure the effect of the collapse on diagram size (Section 6). The budgeted decomposition that keeps the one-time compilation within memory, and the per-sample reading of a maximal reason off the compiled diagrams, are orthogonal and are left out of scope.

The paper is organized as follows. Section 2 fixes the forest, the cell discretization, and the plurality decision. Section 3 develops the margin-absorption accumulator: the counter state, the phased transition relation, the two absorptions and the collapse, and the drop discipline. Section 4 assembles the per-label regions by prioritized peeling and gives the reason-verification test. Section 5 proves the threshold, collapse, and assembly results. Section 6 validates the construction and measures the collapse on tuned forests. Section 7 concludes.

2. Preliminaries

We consider a random forest $\mathcal{F}$ of D decision trees over a finite *feature* set F , classifying into a finite, linearly ordered label set $Y = \{0, 1, \dots, |Y| - 1\}$. Each tree tests the value x_f of a feature f against finitely many constants, $x_f \leq \theta$; the distinct thresholds θ the forest uses on f are its *splits*, collected into the endpoint universe $EU(f)$. Consecutive splits cut the axis of f into $\text{cells}(f) = |EU(f)| + 1$ minimal intervals, the *cells* of f , numbered $0, \dots, \text{cells}(f) - 1$, no two values of which any tree can tell apart, for instance the lowest cell $(-\infty, \theta_{\min(f)}]$ below the smallest split $\theta_{\min(f)}$. A *sample* fixes

one cell per feature: it is the aggregation that a row of a *dataset* induces (each raw value replaced by the cell containing it) and the object the forest votes on. Each tree routes the sample to a leaf, and the label carried by that leaf is the *vote* the tree casts, so the forest is a function of the cells fixed by the sample alone; since these are finitely many, we reason over them rather than over raw values, losing nothing the verdict depends on. The compilation reads the thresholds only through the cells they induce, so it is invariant under any monotone transformation of the feature domain that preserves the ordering of the split thresholds: the passage from a continuous feature space to a finite symbolic one loses nothing the forest can see. We encode the cell index of f with $b_f = \lceil \log_2 \text{cells}(f) \rceil$ Boolean *cell bits* $B_f^0, \dots, B_f^{b_f-1}$, of integer value $X_f = \sum_j 2^j B_f^j$; surplus bit values at or beyond $\text{cells}(f)$ collapse into the last cell. The cell bits of all features form the Boolean variables over which every diagram is ordered. A decision node $x_f \leq \theta$ depends only on the cell, so it is the comparison $X_f \leq \hat{\theta}$ against the index $\hat{\theta}$ of the cell containing θ , a Boolean formula over the cell bits of f of size linear in b_f and with no auxiliary variables. Encoding tree t path by path, the samples on which it votes label ℓ are the union, over the leaves labelled ℓ , of the samples that reach them:

$$Z_t^\ell = \bigvee_{\substack{v \text{ a leaf of } t \\ \text{lab}(v)=\ell}} \left(\bigwedge_{n \text{ on the root-to-}v \text{ path}} \text{lit}_{n,v} \right), \quad (1)$$

where $\text{lit}_{n,v}$ is the comparison $X_f \leq \hat{\theta}$ at the node n (testing feature f) when the path branches true there, and its negation $X_f > \hat{\theta}$ when it branches false. This is the *only* relation in the whole construction that reads the cell bits: it compiles the routing of tree t into a Boolean function of the sample, and from here on the encoding works over the counters and flags alone. As each sample reaches exactly one leaf, the $\{Z_t^\ell\}_{\ell \in Y}$ are mutually exclusive and exhaustive, a partition of the sample space [15, 16].

On a sample c each tree casts one such vote, and we write $\text{votes}_c(\ell)$ for the number of trees that vote ℓ on c , dropping the subscript where the sample is clear from context. The *plurality* (argmax) decision predicts the label with strictly more votes than every other: label i is the *winner* on c when

$$\text{votes}_c(i) > \text{votes}_c(\ell) \quad \text{for every } \ell \in Y \setminus \{i\}, \quad (2)$$

that is, when i is the unique maximum of the tally. A label of maximum vote is a *top label*; the winner, when it exists, is the unique top label. On a sample where the top vote is shared by two or more labels there is no unique winner; we call it a *tie sample*. A continuous reason must certify a label robustly, and a tie sample lies on the boundary where an arbitrarily small change to the input flips the verdict, so tie samples belong to the robust win region of no label and separate them. We agree that a tie carries information, and the construction records it rather than discarding it: a tie sample lands in TIE_i , where i is the first of its top labels in priority order, and the remaining top labels are read off the per-pivot regions compiled in Section 3, as the successors ℓ of i , taken in increasing order, for which the sample lies in $\text{TIE}_\ell^{\text{ig}}$, up to and including the first for which it lies in $\text{WIN}_\ell^{\text{ig}}$. What a tie sample denies is a robust single-label verdict, not the identity of the labels on top. A priority order on Y enters only in Section 4, as a device to decompose the compilation label by label, and plays no part in the decision itself.

For a label i we seek the OBDD of the samples on which i is the plurality winner,

$$\text{WIN}_i = \{c \text{ a sample} : i \text{ is the winner on } c\}, \quad (3)$$

and an auxiliary region TIE_i of tie samples whose shared top labels have i as the smallest index, which is needed to verify reasons (Section 4). These regions are the substrate of region-based explanation. A *box* fixes, for each feature, a contiguous range of its cells, and denotes the set of samples whose cell on every feature lies in that range: the discretized counterpart of an axis-aligned region of input space, and the shape a region-based reason takes. A box that brackets a sample is a robust reason for label i exactly when it lies inside WIN_i , so a compact OBDD for each WIN_i answers reason queries for every sample of a dataset without recompiling. We now construct these diagrams.

3. The Margin-absorption Accumulator

A symbolic fold is expensive because the intermediate relation has to keep every partial tally consistent with the inputs seen so far. The absorption observes that most of that bookkeeping is dead weight: once the pivot leads every rival by more than the number of trees still to be folded, or some rival leads the pivot by that much, no continuation can change the outcome, so the accumulator latches a flag and releases the counters, and all the states decided the same way, whatever tallies produced them, become one state. This is a state abstraction rather than a diagram-level optimization, which is what separates it from variable reordering and dynamic sifting [17]: those shrink a diagram while preserving the function it represents, and we use them too, whereas the collapse changes the relation that gets built, quotienting states no continuation can tell apart. Reordering and the collapse compose rather than compete: both builds of Section 6 run under CUDD [18] with dynamic sift reordering and differ in the collapse alone, so the reduction reported there is measured on top of reordering and not against it.

We fix a *pivot* label $i \in Y$ and compile the contest between i and the labels above it in priority order; the predecessors $0, \dots, i-1$ are handled by the peeling of Section 4. The successors of i are $j_s = i + s$ for $s = 1, \dots, k$ with $k = |Y| - 1 - i$.

Successor s carries a signed *margin* $m_s = \text{votes}(j_s) - \text{votes}(i)$, the lead of rival j_s over the pivot, held biased by the maximal possible lead D (the tree count, since $|m_s| \leq D$) as $u_s = m_s + D \in [0, 2D]$ over $w = \lceil \log_2(2D+1) \rceil$ bits, $u_s = \sum_j 2^j U_s^j$, so that $u_s < D$ means the pivot leads, $u_s > D$ means the rival leads, and $u_s = D$ is a tie. The state adds two latch flags cw, cl (certain-win, certain-lose) and a drop flag d_s per successor:

$$\mathbf{q} = (u_1, \dots, u_k, \text{cw}, \text{cl}, \text{d}_1, \dots, \text{d}_k), \quad \gamma = \text{cw} \vee \text{cl}, \quad (4)$$

where γ abbreviates “the contest is decided.” The sample cells are *static*: every tree reads them to cast its vote, but none changes them, so they are not part of the state and carry no next copy. This is the one departure from the symbolic model-checking discipline, which gives every variable a current and a next copy and prescribes how each evolves; here only the state $\mathbf{q}$, the counters and the flags, advances from one tree to the next, while the sample stays fixed.

Each bit b of $\mathbf{q}$ accordingly has a current copy and a next copy $\bar{b}$, and folding tree t is a relation over these two copies and the static sample, which fixes how the tree votes. Two conventions say how a single bit moves, and we spell them out because the absorption turns on them. A next copy that a relation does not mention is left *free*: the relation constrains it not at all, so once the current copy is quantified away the accumulator no longer depends on that bit. By contrast $[\bar{q} = q]$ abbreviates $\bigwedge_b (\bar{b} \leftrightarrow b)$ and carries every bit forward unchanged: a bit the accumulator pins to a value keeps that value, and a bit the accumulator already leaves free stays free. Writing a relation is thus, bit by bit, a declaration of what is copied forward and what is released, a released bit being exactly a free variable; the release of the counters of a decided contest is what later lets whole states merge. With $\text{clip}(z) = \min(\max(z, 0), 2D)$, folding a relation T into the accumulator A conjoins the two over the shared copies, quantifies away the current copy, and renames the next copy back:

$$\text{step}(A, T) = (\exists \mathbf{q} (A \wedge T)) \{ \bar{\mathbf{q}} \rightarrow \mathbf{q} \}. \quad (5)$$

The initial accumulator, independent of the cell bits, sets every margin to a tie and clears the flags:

$$A_0 = \bigwedge_{s=1}^k (u_s = D) \wedge \neg \text{cw} \wedge \neg \text{cl} \wedge \bigwedge_{s=1}^k \neg \text{d}_s. \quad (6)$$

Phase A folds tree t , moving each non-dropped margin by the vote it casts: a pivot vote lowers every u_s by one, a vote for j_s raises u_s by one, any other label leaves it unchanged. With $\text{inc}_s(\ell) = -1$ if $\ell = i$, $+1$ if $\ell = j_s$, and 0 otherwise,

$$\text{up}_A(\ell) = (\bar{\text{cw}} \leftrightarrow \text{cw}) \wedge (\bar{\text{cl}} \leftrightarrow \text{cl}) \wedge \bigwedge_s [\gamma \vee (\bar{\text{d}}_s \leftrightarrow \text{d}_s)] \wedge \bigwedge_s [\gamma \vee \text{d}_s \vee (\bar{u}_s = \text{clip}(u_s + \text{inc}_s(\ell)))], \quad (7)$$

and the per-tree relation $T_t^A = \bigvee_{\ell \in Y} (Z_t^\ell \wedge \text{up}_A(\ell))$ consults the sample only through Z_t^ℓ (1), which selects the label ℓ the tree votes and so the update $\text{up}_A(\ell)$ that fires; then $A \leftarrow \text{step}(A, T_t^A)$. On a decided (γ) or dropped state the counter is left free, not advanced.

After tree t (the $(t+1)$ -th) is folded, $R = D - (t+1)$ trees remain. Each remaining tree can shift a margin by at most one, so two outcomes can already be certain.

The pivot is sure to strictly beat *every* non-dropped rival, no matter how the remaining trees vote, exactly when every non-dropped margin is below $-R$, i.e. $u_s \leq D - R - 1$. The guard $g_R^{cw} = \bigwedge_s [d_s \vee (u_s \leq D - R - 1)]$ is exactly this condition read off the state, every successor being dropped or already at or below $D - R - 1$, and where it holds the certain-win flag latches, *Phase B*:

$$\tau_R^B = (\gamma \wedge [\bar{q} = q]) \vee (\neg\gamma \wedge g_R^{cw} \wedge \overline{cw} \wedge \neg\overline{cl}) \vee (\neg\gamma \wedge \neg g_R^{cw} \wedge [\bar{q} = q]). \quad (8)$$

The set branch $\overline{cw} \wedge \neg\overline{cl}$ constrains no $\overline{u}_s, \overline{d}_s$, so it leaves them free: this is the collapse.

Symmetrically, some non-dropped rival is sure to stay strictly ahead, so the pivot can no longer even tie, exactly when some non-dropped margin exceeds R , i.e. $u_s \geq D + R + 1$. The guard $g_R^{cl} = \bigvee_s [\neg d_s \wedge (u_s \geq D + R + 1)]$ is exactly this condition, some retained successor already at or above $D + R + 1$, and where it holds the certain-lose flag latches, *Phase C*:

$$\tau_R^C = (\gamma \wedge [\bar{q} = q]) \vee (\neg\gamma \wedge g_R^{cl} \wedge \neg\overline{cw} \wedge \overline{cl}) \vee (\neg\gamma \wedge \neg g_R^{cl} \wedge [\bar{q} = q]). \quad (9)$$

On a certain state there is nothing to carry: the counters are already free. Writing $\hat{A} = \exists u_1 \cdots u_k \exists d_1 \cdots d_k A$ for the abstraction of the counters and drops out of A , the property

$$A \wedge \gamma = \hat{A} \wedge \gamma \quad (10)$$

is an invariant: A_0 has no certain state, the set branches of Phases B and C establish it by leaving every $\overline{u}_s, \overline{d}_s$ free, and every later relation preserves it. Under (10) all decided states merge to one absorbing state per flag, reducing the diagram. We prove the invariant in Section 5.

A non-dropped rival whose margin is so far below the current worst rival that it can never become the binding constraint within R trees may be *dropped*, its counter freed exactly as for a decided state, a second source of collapse. Phase D attaches a biased auxiliary $a = \max(D, \max_{s: \neg d_s} u_s)$, the worst current rival (the pivot constant D guards the case where the pivot leads all),

$$wr(q, a) = \bigvee_{v=D}^{2D} \left[(a = v) \wedge \bigwedge_s (d_s \vee u_s \leq v) \wedge (v = D \vee \bigvee_s (\neg d_s \wedge u_s = v)) \right], \quad A \leftarrow A \wedge (\gamma \vee wr(q, a)). \quad (11)$$

Phase E_s then drops successor s when $a - u_s \geq R + 1$, i.e. s cannot reach the worst rival within the remaining trees, freeing u_s and forcing $\overline{d}_s$; with $dr_{s,R} \equiv (a - u_s \geq R + 1)$ and $[\bar{q} = q \setminus \{u_s, d_s\}]$ the equality on every state bit except u_s (left free) and d_s (set),

$$\tau_R^{E_s} = ((\gamma \vee d_s) \wedge [\bar{q} = q]) \vee (\neg\gamma \wedge \neg d_s \wedge dr_{s,R} \wedge \overline{d}_s \wedge [\bar{q} = q \setminus \{u_s, d_s\}]) \vee (\neg\gamma \wedge \neg d_s \wedge \neg dr_{s,R} \wedge [\bar{q} = q]), \quad (12)$$

and Phase F projects the auxiliary away, $A \leftarrow \exists a A$.

Let $M = \lfloor D/2 \rfloor + 1$. A margin cannot reach $\pm R$ in magnitude before enough trees are folded, so the absorption and drop phases are inert until the midpoint and run only once $t + 1 \geq M$. Algorithm 1 folds the D trees accordingly: Phase A on every tree, and Phases B to F once past the midpoint.

After all D trees, the win and tie regions of the pivot (ignoring predecessors) are read off by projecting away the state:

$$WIN_i^{ig} = \exists q (A_D \wedge cw), \quad TIE_i^{ig} = \exists q (A_D \wedge \neg cw \wedge \neg cl), \quad (13)$$

OBDDs over the cell bits: WIN_i^{ig} is where the pivot strictly beats all successors, TIE_i^{ig} where it ties the top among them. The certain-lose samples fall out of both. For the last label ($k = 0$, no successors) the empty conjunction and disjunction give $WIN_i^{ig} = \top$ and $TIE_i^{ig} = \perp$, so it claims exactly the samples no higher-priority label takes.

Algorithm 1 Folding the D trees into the accumulator for pivot i .

```

1:  $A \leftarrow A_0; \quad M \leftarrow \lfloor D/2 \rfloor + 1$ 
2: for  $t = 0, \dots, D - 1$  do
3:    $R \leftarrow D - (t+1)$ 
4:    $A \leftarrow \text{step}(A, \mathbb{T}_t^A)$  ▷ Phase A: margin update by tree  $t$ 
5:   if  $t + 1 \geq M$  then
6:      $A \leftarrow \text{step}(A, \mathbb{T}_R^B)$  ▷ Phase B: certain-win latch
7:      $A \leftarrow \text{step}(A, \mathbb{T}_R^C)$  ▷ Phase C: certain-lose latch
8:      $A \leftarrow A \wedge (\gamma \vee \text{wr}(\mathbf{q}, a))$  ▷ Phase D: attach worst-rival  $a$ 
9:     for each successor  $s$  do
10:       $A \leftarrow \text{step}(A, \mathbb{T}_R^{E_s})$  ▷ Phase  $E_s$ : drop  $s$  if it can no longer bind
11:     end for
12:      $A \leftarrow \exists a A$  ▷ Phase F: project the auxiliary away
13:   end if
14: end for
15: return  $A_D \leftarrow A$ 

```

4. Assembling the Labels by Prioritized Peeling

The accumulator of Section 3 compiles, for pivot i , the regions $\text{WIN}_i^{\text{ig}}, \text{TIE}_i^{\text{ig}}$ in which i wins or ties the contest among $\{i, i+1, \dots\}$ only, ignoring its predecessors. The true regions follow by peeling, in increasing priority order, the samples already claimed by a higher-priority label:

$$\text{WIN}_i = \text{WIN}_i^{\text{ig}} \setminus \bigcup_{i' < i} (\text{WIN}_{i'} \cup \text{TIE}_{i'}), \quad \text{TIE}_i = \text{TIE}_i^{\text{ig}} \setminus \bigcup_{i' < i} (\text{WIN}_{i'} \cup \text{TIE}_{i'}), \quad (14)$$

a recursion that bottoms out at $i = 0$, where there is no predecessor and $\text{WIN}_0 = \text{WIN}_0^{\text{ig}}, \text{TIE}_0 = \text{TIE}_0^{\text{ig}}$. Each operation is a Boolean combination of OBDDs over the cell bits. Intuitively, if a higher-priority label wins or even ties at the top on a sample, the pivot does not win there; the tie exclusion is essential, since ignoring predecessors lets the pivot spuriously claim a sample where a predecessor ties.

Problem 1 (Reason verification). Given the compiled $\{\text{WIN}_i^{\text{ig}}, \text{TIE}_i^{\text{ig}}\}_i$, a label i , and a box B , decide whether $B \subseteq \text{WIN}_i$, i.e. whether $\mathcal{F}(c) = i$ for every sample $c \in B$.

A box is itself an OBDD over the cell bits: fixing for each feature f a contiguous cell range $[\text{lo}_f, \text{hi}_f]$ (the full range when f is unconstrained), it is the conjunction of the interval comparisons

$$B = \bigwedge_{f \in F} (\text{lo}_f \leq X_f \leq \text{hi}_f), \quad (15)$$

each a Boolean formula over the cell bits of f , just like a decision node. The forest labels every sample of B with i exactly when $B \subseteq \text{WIN}_i$, so by the peeling (14) the reason holds if and only if B lies inside the ignoring-predecessors win of the pivot itself and misses the win and tie of every predecessor, the regions already produced by the bottom-up peeling.

Proposition 1 (Verification test). *A box B satisfies $B \subseteq \text{WIN}_i$ if and only if $B \subseteq \text{WIN}_i^{\text{ig}}$ and $B \cap (\text{WIN}_{i'} \cup \text{TIE}_{i'}) = \emptyset$ for every $i' < i$; equivalently, the interval projection of B is false on the win and tie of every predecessor and true on its own ignoring-predecessors win WIN_i^{ig} .*

Each part is a single Boolean test on the compiled diagrams: a containment $B \subseteq R$ is the emptiness of $B \wedge \neg R$, a disjointness $B \cap R = \emptyset$ the emptiness of $B \wedge R$, each an OBDD conjunction followed by the check whether the result is the $\perp$ diagram. The verifier thus touches only the box (15), the pivot region WIN_i^{ig} , and the already-assembled $\text{WIN}_{i'}, \text{TIE}_{i'}$ of the predecessors, and never re-runs the forest: the compile-once, read-many payoff.

5. Correctness

5.1. Absorption Thresholds

Lemma 1 (Thresholds). *Fix the pivot i and a successor s , and suppose $t + 1$ trees have been folded, so $R = D - (t + 1)$ remain. On a sample with partial margin $m_s = \text{votes}(j_s) - \text{votes}(i)$:*

- (i) *the pivot is guaranteed to end strictly above j_s , regardless of the remaining trees, if and only if $m_s < -R$ (biased: $u_s \leq D - R - 1$);*
- (ii) *the rival j_s is guaranteed to end strictly above the pivot, so the pivot cannot even tie j_s , if and only if $m_s > R$ (biased: $u_s \geq D + R + 1$).*

Both inequalities are strict.

Proof. Each remaining tree changes m_s by -1 (it votes i), $+1$ (it votes j_s), or 0 , so over the R remaining trees the final margin lies in $[m_s - R, m_s + R]$, and every value in this range is attainable by a suitable continuation. For (ii), the most favorable continuation for the pivot drives the margin to its minimum $m_s - R$; the pivot fails to even tie j_s for every continuation iff this minimum is still strictly positive, $m_s - R > 0$, i.e. $m_s > R$. At the boundary $m_s = R$ the minimum is 0 , an attainable tie, which the pivot survives, so the inequality is strict. Case (i) is symmetric, with the worst continuation driving the margin to its maximum $m_s + R$; the pivot strictly beats j_s for every continuation iff $m_s + R < 0$. The biased forms follow from $u_s = m_s + D$. $\square$

Lemma 1 is exactly the guard g_R^{cw} (every non-dropped $m_s < -R$) of Phase B and the guard g_R^{cl} (some non-dropped $m_s > R$) of Phase C. The strictness is what keeps tie samples out of both absorptions: a sample that can still be forced to a top tie is neither a certain win nor a certain loss.

5.2. The Collapse Invariant

Lemma 2 (Collapse). *Identity (10) holds after every phase: on a certain state, the accumulator does not depend on the counters u_s or the drop flags d_s . Consequently the certain branch $\gamma \wedge [\bar{q} = q]$ of Phases B, C, and E is interchangeable, after step, with the flags-only collapse $\gamma \wedge (\bar{cw} \leftrightarrow cw) \wedge (\bar{cl} \leftrightarrow cl)$.*

Proof. By induction over the phases. A_0 has no certain state, so the claim holds vacuously. A state becomes certain only in the set branch of Phase B ((8), $\bar{cw} \wedge \neg \bar{cl}$) or Phase C ((9), $\neg \bar{cw} \wedge \bar{cl}$); neither branch constrains any $\bar{u}_s$ or $\bar{d}_s$, so by the omitted-variable convention these are free in the next state, establishing the invariant at the moment of becoming certain. Every later relation preserves it: Phase A frees the counters on γ through its $\gamma \vee (\dots)$ guards (7); Phase D conjoins $\gamma \vee wr$, true on γ for any counters; Phase E takes its $[\bar{q} = q]$ branch on a certain state, copying each next counter from an already-free current counter and so preserving freeness; and a full equality $[\bar{q} = q]$ likewise copies an already-free bit to a free bit. For the interchangeability, step abstracts the current copy and renames, so both forms copy the flags faithfully via $\bar{cw} \leftrightarrow cw$ and $\bar{cl} \leftrightarrow cl$; under the invariant the current u_s, d_s of a certain state range over their whole domain, so equality emits every $\bar{u}_s, \bar{d}_s$, the same set the omission leaves free. $\square$

5.3. The Assembled Regions Are the Argmax of the Forest

Lemma 3 (Per-pivot closure). *For every sample c , $c \in \text{WIN}_i^{\text{ig}}$ iff the pivot i strictly beats every successor on c , i.e. $\text{votes}(i) > \text{votes}(j)$ for all $j > i$; and $c \in \text{TIE}_i^{\text{ig}}$ iff $\text{votes}(i) \geq \text{votes}(j)$ for all $j > i$ with equality for some such j .*

Proof. At $R = 0$ the guards of Lemma 1 read: certain-win iff every non-dropped $m_s < 0$, and certain-lose iff some non-dropped $m_s > 0$. We show the closure over the non-dropped successors reads the same verdict as the full set, by showing the maximum final margin $m^* = \max_s m_s$ is attained by a non-dropped successor whenever $m^* \geq 0$. A rival s is dropped at the step with R' trees remaining

when $a - u_s \geq R' + 1$, where $a = \max(D, \max_{r \text{ retained}} u_r)$. Its *witness* is the pivot when $a = D$ and otherwise a retained rival r with $u_r = a$; unbiaseding, the gap from the witness down to s is at least $R' + 1$ at the drop. Throughout, m_s is the true running margin $\text{votes}(j_s) - \text{votes}(i)$, which keeps evolving after the counter u_s is frozen at the drop; freezing only removes s from the closure, not from the tally. Among the R' remaining trees only a j_s vote narrows the true gap, by one (a pivot vote lowers every margin, hence both ends of the gap; a witness vote widens it; any other leaves it unchanged), so the final gap is at least 1: the witness ends strictly above s . Now suppose $m^* \geq 0$ were attained by a dropped s . Its witness cannot be the pivot, which would force $m_s \leq -1$; so the witness is a rival ending strictly above m^* , contradicting maximality unless it too is dropped, whose own witness then ends higher still. As there are finitely many successors, this strictly increasing chain is impossible, so m^* is attained by a non-dropped successor. Hence the closure detects a loss ($m^* > 0$) or a top tie ($m^* = 0$) exactly when it truly occurs, and reads certain-win exactly when every $m_s < 0$. So cw holds at A_D iff $\text{votes}(j_s) < \text{votes}(i)$ for all s , and $\neg \text{cw} \wedge \neg \text{cl}$ iff all $m_s \leq 0$ with some $m_s = 0$. Equations (13) project these onto the cell bits. $\square$

Theorem 1 (Correctness). *For every sample c and label i , $c \in \text{WIN}_i$ as defined by the peeling (14) if and only if i is the winner on c , i.e. $\text{votes}(i) > \text{votes}(\ell)$ for all $\ell \neq i$. The regions $\{\text{WIN}_i\}_{i \in Y}$ are pairwise disjoint, and their union is exactly the samples that have a unique winner; the remaining tie samples lie in no WIN_i .*

Proof. Fix a sample c and let $T = \{\ell \in Y : \text{votes}(\ell) \geq \text{votes}(\ell') \text{ for all } \ell'\}$ be its set of top labels, so i is the winner iff $T = \{i\}$. By Lemma 3, WIN_i^{ig} is the set of samples with $\text{votes}(i) > \text{votes}(j)$ for all successors $j > i$, and TIE_i^{ig} the set with $\text{votes}(i) \geq \text{votes}(j)$ for all $j > i$ and equality for some such j . We prove by strong induction on i that

$$\text{WIN}_i = \{c : T = \{i\}\}, \quad \text{TIE}_i = \{c : i = \min T \text{ and } |T| \geq 2\}.$$

By the induction hypothesis $\text{WIN}_{i'} \cup \text{TIE}_{i'} = \{c : i' = \min T\}$ for each $i' < i$, so the subtracted set in (14) is $P_i = \bigcup_{i' < i} (\text{WIN}_{i'} \cup \text{TIE}_{i'}) = \{c : \min T < i\}$, the samples some predecessor of i tops. At the base $i = 0$ this union is empty, $P_0 = \emptyset$, and the identities reduce to $\text{WIN}_0 = \text{WIN}_0^{\text{ig}}$, $\text{TIE}_0 = \text{TIE}_0^{\text{ig}}$.

For $\text{WIN}_i = \text{WIN}_i^{\text{ig}} \setminus P_i$: a sample of WIN_i^{ig} has i strictly above every successor. If $\min T < i$ it is removed by P_i ; otherwise no predecessor is a top label, and since i already beats every successor, i is the unique maximum, $T = \{i\}$. Conversely $T = \{i\}$ puts i strictly above all labels, so $c \in \text{WIN}_i^{\text{ig}}$ and $\min T = i \notin P_i$. Hence $\text{WIN}_i = \{c : T = \{i\}\}$. For $\text{TIE}_i = \text{TIE}_i^{\text{ig}} \setminus P_i$: a sample of TIE_i^{ig} has i tying the top among $\{i, i+1, \dots\}$, so $i \in T$ and $|T| \geq 2$; it survives P_i iff $\min T \geq i$, i.e. $\min T = i$. Conversely $i = \min T$ with $|T| \geq 2$ puts i at the shared top among its successors, so $c \in \text{TIE}_i^{\text{ig}}$ and $\min T = i \notin P_i$.

Finally the $\text{WIN}_i = \{c : T = \{i\}\}$ are disjoint and their union is the samples with $|T| = 1$; every tie sample has $|T| \geq 2$ and lies in $\text{TIE}_{\min T}$, in no WIN_i . $\square$

Proposition 1 is the box form of (14): a box is a set of samples, so $B \subseteq \text{WIN}_i$ iff $B \subseteq \text{WIN}_i^{\text{ig}}$ and $B \cap P_i = \emptyset$, i.e. B misses the win and tie of every predecessor.

Example 1. Let $D = 5$ trees over one feature with cells $c_0, \dots, c_4$, which here are the samples, and labels $\{0, 1, 2\}$, and suppose the per-cell tallies make 0 the unique winner on c_0 , the top shared by $\{0, 1\}$ on c_1 , label 1 the winner on c_2, c_3 , and 2 on c_4 (Figure 1). Absorption starts at the midpoint $M = \lfloor 5/2 \rfloor + 1 = 3$. For pivot 0, the margins $m_s = \text{votes}(j_s) - \text{votes}(0)$ to rivals 1, 2 end at $(-3, -4)$ on c_0 (certain-win: both already below $-R$ when reached), at $(0, -1)$ on c_1 (a top tie with 1, every $m_s \leq 0$ with $m_1 = 0$, so $\neg \text{cw} \wedge \neg \text{cl}$), and trigger certain-lose, some $m_s > R$, on c_2, c_3, c_4 . So $\text{WIN}_0^{\text{ig}} = \text{WIN}_0 = \{c_0\}$ and $\text{TIE}_0 = \{c_1\}$. Pivot 1, ignoring 0, strictly beats rival 2 on $\{c_0, c_1, c_2, c_3\}$; peeling $\text{WIN}_0 \cup \text{TIE}_0 = \{c_0, c_1\}$ leaves $\text{WIN}_1 = \{c_2, c_3\}$. Pivot 2 peels to $\text{WIN}_2 = \{c_4\}$. These coincide with the argmax of the forest, the tie sample c_1 lies in no WIN_i , and the box $\{c_2, c_3\}$ is a valid reason for label 1 whereas $\{c_1, c_2\}$ is not, since it meets TIE_0 .

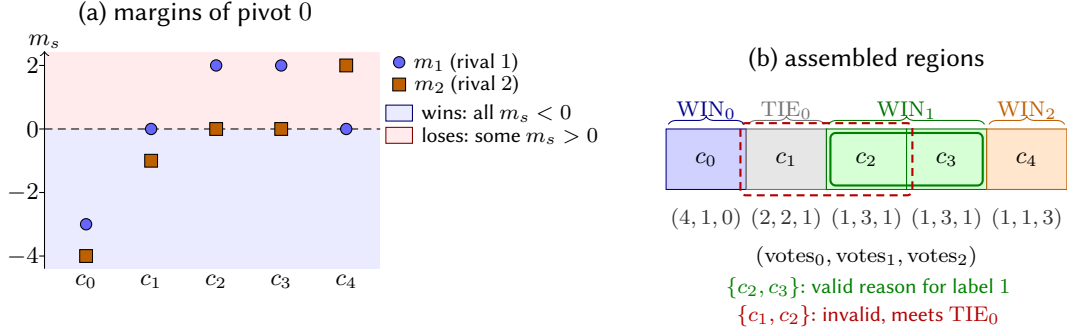


Figure 1: The run of Example 1. **(a)** the final signed margins m_1, m_2 of pivot 0, against the win band ($m_s < 0$) and lose band ($m_s > 0$); **(b)** the regions the prioritized peeling assembles from the per-cell tallies, with one valid and one invalid candidate reason for label 1.

6. Experimental Validation

The overhead is confined to the trees past the midpoint, where the two latch phases add one relational product each beyond the margin update of Phase A, and to the two flag bits added to the state. Because every product that follows a latch acts on a smaller intermediate diagram, the cost is recovered inside the same build, and the timings at the end of this section report the net effect.

We implemented the accumulator and measured the effect of the absorption across the classification datasets of the Penn Machine Learning Benchmarks (PMLB), in ascending order of feature count.¹ To isolate the core mechanism of the absorption, the *collapse*, we build the reduced WIN/TIE on the same accumulator twice, toggling only the collapse and holding the drop discipline (Phases D to F) off throughout; the drop is an optional refinement assessed separately below. With the collapse off, a decided state copies its counters forward instead of freeing them, which by Lemma 2 computes the identical relation, so the two builds also serve as a built-in correctness control.

A single seed (1234) governs every stochastic step, the split, the cross-validation partitions, the Optuna sampler, and each forest fit, so the whole pipeline is deterministic and a single run reproduces it exactly, with no replication needed. The seed acts on the input to the compilation, that is, on which forest is trained, and not on the compilation itself, which is deterministic and exact for whatever forest it receives; re-seeding would compile other forests rather than exercise a different path. Per dataset we draw a stratified 80/20 train/test split and select the forest on the train split only. We first tune the tree depth and the minimum leaf size with Optuna (15 trials, Tree-structured Parzen Estimator sampler) over $\text{depth} \in [3, 10]$ and $\text{min_leaf} \in [1, 8]$, maximizing the mean accuracy of a 5-fold stratified cross-validation at a fixed 40 trees. Holding the tuned depth and leaf, we then fix the tree count by the one-standard-error rule over the grid $\{10, 20, 30, 40, 60, 80, 120, 160, 240\}$, taking the smallest forest whose mean 5-fold cross-validation accuracy lies within one standard error (the sample standard deviation of the fold accuracies over $\sqrt{5}$) of the best grid point, the parsimonious sufficient forest, and refit it on the full train split. For each forest we record the trees D , the features f , and the splits (distinct thresholds summed over the used features), and build the relation under CUDD 3.0.0 with dynamic sift reordering. The response is the peak intermediate diagram, the maximum accumulator DagSize as the trees are folded in, reported with the collapse on and off and as their ratio (no-absorption peak over collapse peak). We measure it for pivot 0, the pivot with the most successors and so the largest state, whose peak is a faithful proxy for the peak over all pivots and whose cost is independent of the label count. CUDD enforces a high node budget ($3.2 \cdot 10^7$ live nodes, via `Cudd_SetMaxLive`) and a per-build time limit (90 s, via `Cudd_SetTimeLimit`), chosen so both builds complete on most datasets; a build that exceeds either is aborted and the dataset marked infeasible. We sweep in feature order and stop after 12 consecutive datasets on which the collapse build fails, the point we take as the feasibility

¹The code (the native accumulator and the experiment harness) and the recorded sweep are available for reproduction at <https://github.com/PietroSala/margin-absorption>.

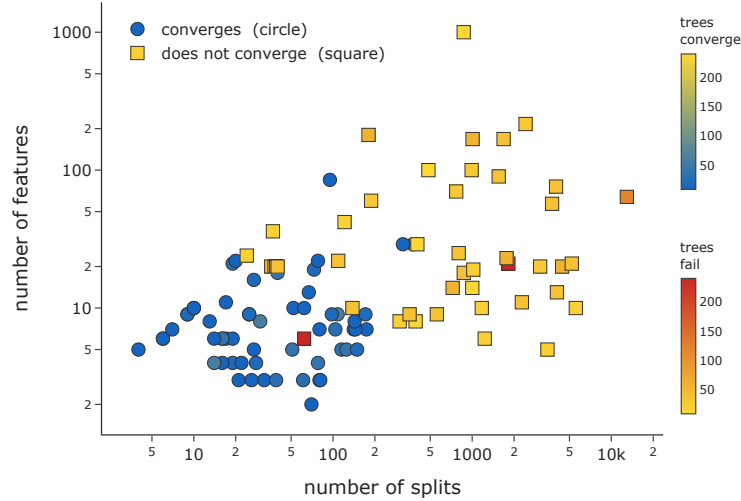


Figure 2: Feasibility by split count and feature count (both log): converging forests are circles shaded blue (few trees) to yellow (many), non-converging forests squares shaded yellow to red. The two axes and the tree colour carry all three forest-size drivers; the feasible region sits at low splits, features, and trees, and a forest leaves it as any of them grows, with the labels a fourth driver (the red square among the circles is `krkopt`, infeasible at low splits and features through its 240 trees and 18 labels). The same datasets are infeasible with the collapse and without it, so the collapse lowers the peak without moving this frontier.

frontier.

On every dataset the builds with and without the collapse produced bit-identical WIN diagrams, an empirical corroboration of the collapse invariant (Lemma 2) and the per-pivot closure (Lemma 3) across the benchmark.

The same growth bounds what compiles at all. Figure 2 plots every dataset by its split and feature counts and separates the forests the collapse compiles from those it does not: the feasible region sits at low splits and features, and a forest crosses into infeasibility as either grows, or as its labels or trees do, high splits (`phoneme`, `magic`), many labels (`krkopt`), or, even at modest split counts, many features (`dna` at 180, `1ed24` at 24), the last through the interactions a wide feature set breeds. Tellingly, the same 45 datasets fail with the collapse and without it: the collapse shrinks the peak (Table 1) but does not move this frontier, so on these forests the budgeted decomposition (Section 7) is required. We thus report the collapse as a verified reduction, a small factor on small forests growing to several-fold on dense ones, rather than an established widening of the feasible region. The practical content of that reduction is headroom: what a node budget has to hold during the one-time compilation is the peak intermediate diagram, and the collapse lowers it by a median factor of 1.46 over the datasets of Table 1. The budgeted decomposition of Section 7 buys that headroom by case splitting, so a peak lower by this factor is headroom it does not have to buy.

The absorption also costs nothing to run. In a separate run timing all three configurations on sixteen datasets, ten of which complete every configuration, the build with the collapse is faster than the build without it on nine, by a median factor of $1.15\times$ and by up to $5.5\times$, and slower on one by 4%: the cost of the package follows the size of the intermediate diagrams it manipulates, so releasing the counters of a decided state removes work rather than adding it. The drop of Phases D to F behaves differently, costing a median $6.1\times$ in build time over the same forests without lowering the peak on any of them, so we hold it off throughout, attribute the reduction entirely to the collapse, and treat the drop as an optional refinement.

Table 1: The PMLB classification datasets whose forest the collapse compiles within the budget, sorted by split count: peak intermediate diagram with the absorption collapse on and off, their ratio (no absorption over collapse), and the size of the compiled WIN (bit-identical with the collapse on or off, far below either peak). f = features, $|Y|$ = labels, splits = distinct thresholds, D = trees. The datasets that do not compile appear in Figure 2. Names over 24 characters are truncated.

dataset	f	$ Y $	splits	D	collapse	no abs.	ratio	final
parity5	5	2	4	10	13	19	$1.46\times$	3
corral	6	2	6	120	199	199	$1.0\times$	5
mux6	6	2	6	10	57	72	$1.26\times$	10
led7	7	10	7	10	2507	3218	$1.28\times$	16
threeOf9	9	2	9	40	471	543	$1.15\times$	26
xd6	9	2	9	20	298	336	$1.13\times$	16
mofn_3_7_10	10	2	10	60	702	848	$1.21\times$	36
parity5+5	10	2	10	10	481	616	$1.28\times$	182
postoperative_patient_d...	8	2	13	10	47	51	$1.09\times$	1
hayes_roth	4	3	14	60	775	875	$1.13\times$	31
monk3	6	2	14	10	196	234	$1.19\times$	30
balance_scale	4	3	16	10	429	528	$1.23\times$	1
monk1	6	2	16	120	1241	1438	$1.16\times$	45
monk2	6	2	16	40	731	927	$1.27\times$	73
analcata_data_bankruptcy	6	2	17	10	69	148	$2.14\times$	49
analcata_data_fraud	11	2	17	10	234	490	$2.09\times$	128
analcata_data_dmft	4	6	19	20	3868	4791	$1.24\times$	48
analcata_data_creditscore	6	2	19	10	47	110	$2.34\times$	13
car_evaluation	21	4	19	30	4606	5182	$1.13\times$	70
spect	22	2	20	10	124	166	$1.34\times$	14
analcata_data_asbestos	3	2	21	10	47	69	$1.47\times$	9
iris	4	3	22	10	167	205	$1.23\times$	16
breast_cancer	9	2	25	20	884	1295	$1.46\times$	337
tic_tac_toe	9	2	25	20	15302	17851	$1.17\times$	1421
analcata_data_happiness	3	3	26	10	385	415	$1.08\times$	56
irish	5	2	27	10	173	277	$1.6\times$	68
labor	16	2	27	20	654	1123	$1.72\times$	413
analcata_data_aids	4	2	28	20	188	352	$1.87\times$	54
nursery	8	4	30	60	25442	25769	$1.01\times$	3
analcata_data_boxing1	3	2	32	10	222	304	$1.37\times$	73
analcata_data_boxing2	3	2	39	30	452	516	$1.14\times$	57
lymphography	18	4	40	40	95510	196792	$2.06\times$	1
analcata_data_germangss	5	4	51	40	5008	7369	$1.47\times$	164
analcata_data_cyyoung8092	10	2	52	10	1895	4213	$2.22\times$	1074
haberman	3	2	61	30	1014	1168	$1.15\times$	130
analcata_data_cyyoung9302	10	2	62	10	9881	11092	$1.12\times$	5690
wine_recognition	13	3	67	10	15429	45427	$2.94\times$	12550
prnn_synth	2	2	70	10	205	331	$1.61\times$	52
hepatitis	19	2	73	20	2791	7364	$2.64\times$	1042
analcata_data_lawsuit	4	2	78	40	1490	3128	$2.1\times$	307
agaricus_lepiota	22	2	78	10	1516598	2178358	$1.44\times$	535962
confidence	3	6	80	10	5582	11855	$2.12\times$	460
penguins	7	3	80	10	6114	11247	$1.84\times$	3589
lupus	3	2	81	10	1142	1774	$1.55\times$	463
coil2000	85	2	95	10	808	633	$0.78\times$	657
saheart	9	2	98	20	15880	20970	$1.32\times$	6462
appendicitis	7	2	104	30	5315	21989	$4.14\times$	3926
analcata_data_japansolvent	9	2	108	60	4574	37424	$8.18\times$	1554
bupa	5	2	115	30	21348	57290	$2.68\times$	5477
new_thyroid	5	3	125	40	7791	18402	$2.36\times$	901
ecoli	7	5	141	20	75262	163779	$2.18\times$	4621
biomed	8	2	143	10	303229	424964	$1.4\times$	111033
prnn_crabs	7	2	145	10	42798	152921	$3.57\times$	32392
tae	5	3	149	20	29543	53840	$1.82\times$	2430
profb	9	2	171	20	135946	355544	$2.62\times$	79875

continued on next page

Table 1, continued

dataset	f	$ Y $	splits	D	collapse	no abs.	ratio	final
cloud	7	4	174	10	440238	965693	$2.19\times$	206805
allhyper	29	4	318	10	2193889	6263871	$2.86\times$	148632
allbp	29	3	375	20	1180036	3452008	$2.93\times$	1

7. Conclusion

We have presented a relational encoding that compiles the plurality-winner regions of a random forest into OBDDs by folding the trees through per-pivot signed margin counters, with two absorption flags that latch a decided contest and collapse all decided states to one, and a drop discipline that frees a counter that can no longer bind. A prioritized one-vs-rest peeling assembles the per-label win and tie regions and gives a sound and complete reason-verification test. We proved the exact absorption thresholds $m_s < -R$ and $m_s > R$, both strict at the tie boundary, a collapse invariant showing a decided state is independent of its counters, and that the assembled win regions are the argmax of the forest. Although the construction is presented for plurality voting, its underlying ideas are more general: the cell abstraction and the margin-state encoding could apply to any voting rule defined by comparisons between counts, while the absorption thresholds adapt to the decision criterion.

The encoding is the compilation core of a compile-once, read-many explainer. Such a representation is particularly attractive in applications where explanations must be produced under tight latency constraints, Electroencephalography (EEG)-based brain computer interfaces being one example [19]. Moreover, the compiled regions naturally support analyses beyond explanation, including robustness measures [20]. Two components complete that pipeline and are left to future work: a budgeted tableau decomposition that case-splits a single cell bit when the intermediate diagram exceeds a node budget, which Section 6 shows is needed already at moderate size on the densest forests, those with many splits or labels; and the per-sample reading of a coordinate-wise maximal robust box off the compiled win and tie diagrams. Studying the collapse *within* the budgeted decomposition on large forests, and a comparison with per-sample formal baselines such as maxiAXP [9, 21], are the natural next steps.

Acknowledgments

This work has been carried out while Emanuele Chini was enrolled in the Italian National Doctorate on Artificial Intelligence run by Sapienza University of Rome in collaboration with the University of Verona.

Declaration on Generative AI

During the preparation of this work, the authors used an AI-based assistant solely in order to improve the English language of the manuscript. Everything else is entirely the work of the authors. The tool used was Claude Opus 4.8, a large language model developed by Anthropic.

References

- [1] L. Breiman, Random forests, *Machine learning* 45 (2001) 5–32.
- [2] M. Khalilia, S. Chakraborty, M. Popescu, Predicting disease risks from highly imbalanced data using random forest, *BMC Medical Informatics and Decision Making* 11 (2011) 51. doi:10.1186/1472-6947-11-51.
- [3] X. Zhang, Y. Yang, Z. Zhou, A novel credit scoring model based on optimized random forest, in: 2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC), IEEE, 2018, pp. 60–65. doi:10.1109/CCWC.2018.8301707.

- [4] S. Lundberg, S.-I. Lee, A unified approach to interpreting model predictions, 2017. URL: <https://arxiv.org/abs/1705.07874>. arXiv:1705.07874.
- [5] M. T. Ribeiro, S. Singh, C. Guestrin, "why should i trust you?": Explaining the predictions of any classifier, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '16, Association for Computing Machinery, New York, NY, USA, 2016, p. 1135–1144. URL: <https://doi.org/10.1145/2939672.2939778>. doi:10.1145/2939672.2939778.
- [6] M. T. Ribeiro, S. Singh, C. Guestrin, Anchors: High-precision model-agnostic explanations, in: Proceedings of the AAAI conference on artificial intelligence, volume 32, 2018.
- [7] Y. Izza, J. Marques-Silva, On explaining random forests with sat, 2021. URL: <https://arxiv.org/abs/2105.10278>. arXiv:2105.10278.
- [8] A. Ignatiev, N. Narodytska, J. Marques-Silva, Abduction-based explanations for machine learning models, in: Proceedings of the AAAI conference on artificial intelligence, volume 33, 2019, pp. 1511–1519.
- [9] Y. Izza, A. Ignatiev, S. Rubin, J. Marques-Silva, P. J. Stuckey, Most general explanations of tree ensembles, in: Proceedings of the 34th International Joint Conference on Artificial Intelligence (IJCAI), 2025.
- [10] A. Darwiche, A. Hirth, On the reasons behind decisions, in: ECAI 2020, IOS Press, 2020, pp. 712–720.
- [11] A. Darwiche, A. Hirth, On the (complete) reasons behind decisions, Journal of Logic, Language and Information 32 (2022).
- [12] R. E. Bryant, Graph-based algorithms for boolean function manipulation, Computers, IEEE Transactions on 100 (1986) 677–691.
- [13] J. R. Burch, E. M. Clarke, K. L. McMillan, D. L. Dill, L. J. Hwang, Symbolic model checking: 10^{20} states and beyond, Information and Computation 98 (1992) 142–170. doi:10.1016/0890-5401(92)90017-A.
- [14] K. L. McMillan, Symbolic Model Checking, Kluwer Academic Publishers, Boston, 1993. doi:10.1007/978-1-4615-3190-6.
- [15] A. Choi, A. Shih, A. Goyanka, A. Darwiche, On symbolically encoding the behavior of random forests, CoRR abs/2007.01493 (2020). URL: <https://arxiv.org/abs/2007.01493>. arXiv:2007.01493.
- [16] G. Audemard, S. Bellart, L. Bounia, F. Koriche, J.-M. Lagniez, P. Marquis, On the explanatory power of decision trees, 2021. URL: <https://arxiv.org/abs/2108.05266>. arXiv:2108.05266.
- [17] R. Rudell, Dynamic variable ordering for ordered binary decision diagrams, in: Proceedings of the IEEE/ACM International Conference on Computer-Aided Design (ICCAD), 1993, pp. 42–47.
- [18] F. Somenzi, Cudd: Cu decision diagram package release 2.3. 0, University of Colorado at Boulder 621 (1998) 2.
- [19] R. Portillo-Lara, B. Tahirbegi, C. A. Chapman, J. A. Goding, R. A. Green, Mind the gap: State-of-the-art technologies and applications for eeg-based brain–computer interfaces, APL bioengineering 5 (2021).
- [20] D. Amadori, E. Chini, P. Sala, Drifts: Distributed robustness for time series is feasible, distributed, and fun, in: 2026 IEEE Conference on Artificial Intelligence (CAI), 2026, pp. 114–121. doi:10.1109/CAI68641.2026.11536523.
- [21] Y. Izza, A. Ignatiev, P. J. Stuckey, J. Marques-Silva, Delivering inflated explanations, in: Proceedings of the 38th AAAI Conference on Artificial Intelligence, 2024.