

Original Article

SCMI-TREE: SYNERGY-AWARE DECISION TREES VIA CONDITIONAL MUTUAL INFORMATION WITH ADAPTIVE LOOKAHEAD

Taha A. Ababakar^{1,*} , Hakar J. M. Salih¹ , Dilheem H. Sabri² , and Omar M. Ahmed^{3,*} 

¹Department of Computer Science, College of Science, University of Zakho, Zakho, Duhok, Kurdistan Region, Iraq.

²Department of Mathematics, College of Science, University of Zakho, Zakho, Duhok, Kurdistan Region, Iraq.

³Computer Information System Department, Technical College of Zakho, Duhok Polytechnic University, Duhok, Kurdistan Region, Iraq.

*Corresponding author; E-mail: omar.alzakholi@dpu.edu.krd (Tel: +964-7504711356)

ABSTRACT

Received
July 22, 2026

Accepted
September 19, 2026

Published
October 10, 2026

Decision trees built with greedy axis-aligned splits perform well when individual features carry a meaningful signal about the target. They fail, often badly, in the setting where the useful information resides in the joint behavior of two or more features rather than in any single one. The exclusive-or problem is the textbook example, since no single feature predicts the label, but the pair does. This study presents SCMI-Tree, a tree induction algorithm whose splitting score augments mutual information with two additional terms, namely a synergy bonus for features that interact constructively with the ancestor splits along the path, and a redundancy penalty for features that overlap with them. A dataset-level synergy score is computed once at fit time from a random sample of feature pairs, which allows the algorithm to decide whether to engage a lookahead mechanism that addresses the root exclusive-or problem, and to skip that overhead when it would not help. SCMI-Tree was evaluated on eight classification datasets comprising four real benchmarks and four synthetic tasks of varying difficulty. The lookahead variant achieved the best average rank of 1.88 across the eight datasets, outperforming both Gini-based CART at 4.12 and entropy-based ID3 at 2.75. On a noisy exclusive-or task with eight irrelevant features the difference was substantial, reaching 89.3% accuracy for SCMI-Tree against 53.2% for CART. Ablation experiments show that two failure modes of conventional greedy induction, namely sensitivity to noise features and dependence on tree depth, are largely avoided by the proposed method.

Keywords: Decision Trees, Feature Synergy, Interaction Information, Interpretable Machine Learning, Lookahead.

1. INTRODUCTION

Decision trees remain in widespread use. Despite the prevalence of deep learning, tabular problems continue to rely heavily on them for several reasons: training is fast; tuning requirements are minimal; and a tree of twenty or thirty nodes can be inspected directly by a domain expert. They also serve as the base learners for the random forest and gradient-boosting ensembles that dominate tabular benchmarks, so that the quality of the individual trees remains important even when the ensemble produces the final prediction.

Most induction algorithms, such as CART (Breiman *et al.*, 1984) and ID3 and its descendants C4.5 and C5.0 (Quinlan, 1986, 1993), work by comparing features one by one and choosing the one with the highest impurity reduction, with Gini impurity and entropy being the standard choices of impurity measure. This is a greedy, axis-aligned, marginal algorithm: features are independently assessed for their ability to predict the target on a local subset of the data at a tree node. This strategy is sufficient for a good many problems, but not for an important class. When the target depends on a joint configuration of features rather than on any single feature, a marginal greedy splitter has no signal to work with at the root. The exclusive-or (XOR) problem is the canonical case, but the phenomenon is not just an academic curiosity. Feature interactions appear in real data, for example in some biomedical settings where a gene-gene interaction matters but neither gene alone does, in fraud detection where a pattern emerges only from a

combination of variables, and in any problem derived from combinatorial structure. The greedy tree confronted with these data is either going to split almost arbitrarily at the root or simply stop, because it does not know which splits are useful in this case.

Information theory has a precise vocabulary for feature interactions. The mutual information $I(X; Y)$ measures how much knowing X reduces uncertainty about Y . The joint mutual information $I((X_1, X_2); Y)$ measures the same quantity for a pair of features. The interaction information $II(X_1; X_2; Y)$, introduced by McGill (1954), is the difference between the two. It is positive when the features overlap in what they tell us about the target, a situation described as redundancy, and negative when the pair carries more information than the sum of its parts, a situation described as synergy. XOR has strongly negative interaction information because the joint distribution is fully informative, while neither marginal distribution provides any information. Interaction information has been used for feature selection (Brown *et al.*, 2012; Vergara & Estévez, 2014; Beraha *et al.*, 2019) and in studies of neural representation (Tax *et al.*, 2017), but not, as far as the authors are aware, as a built-in component of decision tree induction. Interaction information also underlies the analysis of synergy among interacting genes in computational biology (Anastassiou, 2007), where the joint effect of several variants on a phenotype cannot be recovered from any single variant in isolation. Figure 1 illustrates this concretely: panel (a) shows the exclusive-or arrangement in which

Access this article online



DOI: <https://doi.org/10.25271/sjuoz.2026.14.4.1983>

Printed ISSN 2410-7549;
Electronic ISSN 2414-6943

Science Journal of University of Zakho
Vol. 14, No. 04, pp. 619-630, October -2026

This is an open access article under a CC BY-NC-SA 4.0 license
(<https://creativecommons.org/licenses/by-nc-sa/4.0/>)

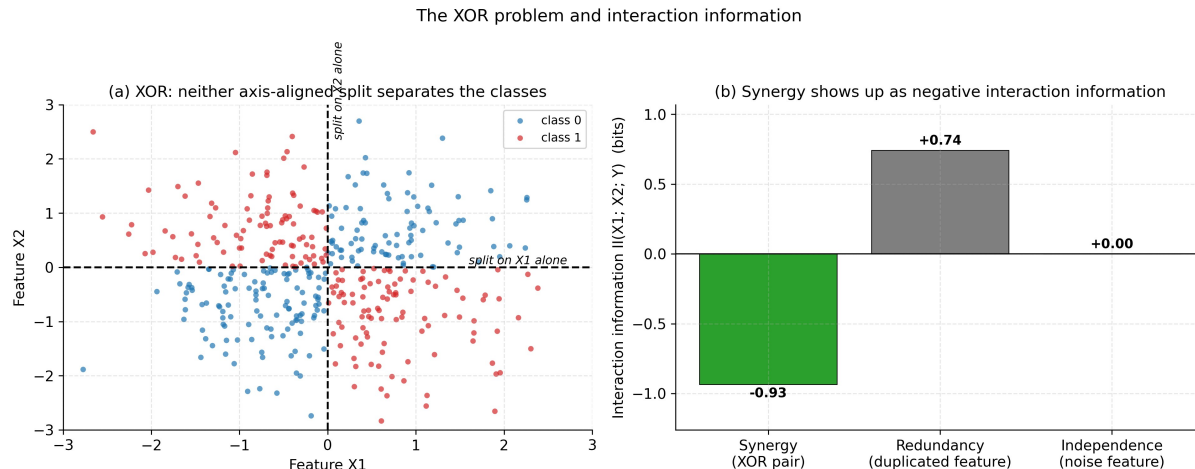


Figure 1: Illustration of feature synergy. Panel (a) shows the exclusive-or arrangement, in which the two classes cannot be separated by any single axis-aligned split. Panel (b) shows the interaction information of a synergistic (XOR) pair, a redundant pair and an independent pair; synergy corresponds to negative interaction information.

no single axis-aligned split separates the classes, while panel (b) contrasts the interaction information of a synergistic pair, a redundant pair and an independent pair, making explicit that synergy corresponds to negative interaction information.

Splitting Criteria for Decision Trees:

Gini impurity and entropy are the two dominant choices. In practice, the two often agree on which split is best (Raileanu & Stoffel, 2004), and the choice between them is more often a matter of software convention rather than principle. Both share the same limitation that motivates this study, in that each evaluates features in isolation against the target, with no mechanism for considering relationships between features. C4.5 introduced gain ratio (Quinlan, 1993) to handle the bias toward high-cardinality features, but this addresses a different issue.

Other splitting criteria have appeared over the years for more specialized use cases. For class-imbalanced settings, Hellinger-based trees (Cieslak *et al.*, 2012; Lyon *et al.*, 2014; Akash *et al.*, 2019) trade Gini for the skew-insensitive Hellinger distance. Nonlinear target dependencies have been addressed by way of distance correlation (Tan *et al.*, 2022). All of these, however, share the same architectural blindness as Gini and entropy, since each feature is judged on its own and never against any other.

Trees That Look Beyond Axis-Aligned Splits:

Oblique trees result when this constraint is relaxed. A widely used algorithm is OC1 (Murthy *et al.*, 1994), with more recent work using evolutionary search (Wickramarachchi *et al.*, 2016), gradient descent for soft trees (Norouzi *et al.*, 2015), and differentiable relaxations (Karhikeyan *et al.*, 2022). In principle, XOR can be learned by a single hyperplane defined by $x_1 - x_2$, provided that the optimizer converges to this solution; however, this comes at the cost of interpretability. A physician can immediately understand a rule of the form "is feature 3 lower than 2.5", but cannot readily interpret a weighted sum of eight features with a threshold of 1.4.

Lookahead splitting goes back considerably earlier to Elomaa and Malinen (2003) and Esmeir and Markovitch (2007), who evaluated each candidate split by how well the resulting children could themselves be split. This is not a standard technique, however, for one main reason, which is cost. A lookahead of one level increases per-node costs by a factor of the number of features, and a lookahead of two levels increases them by the square, so a lookahead of three levels is generally not feasible at all. The proposal advanced here is to apply lookahead selectively, only when cheap criterion indicates that it is worth doing.

Ensembles and the Interpretability Trade-off:

Comparison with ensembles is unavoidable. Random forests (Breiman, 2001) and boosted trees (Chen & Guestrin, 2016; Ke *et al.*, 2017) outperform single trees on tabular data, and the consensus on that point has been strong, with Grinsztajn *et al.* (2022) offering another confirmation. Feature-importance charts recover some of the interpretability that an ensemble sacrifices, but they do not provide an alternative to the interpretability a shallow tree offers, namely a path that can be read from start to finish.

In contexts where interpretability is crucial, such as clinical decision support and credit risk, and in the broader category of high-stakes and regulated environments (Rudin, 2019), the question is not whether an ensemble can achieve better accuracy but whether being un-auditable is acceptable. In most such cases it is not. The present work therefore operates within the realm of single-tree models, performs no comparisons with forests or boosting models, and makes no claim to outperform them.

Ensembles and the Interpretability Trade-off:

None of the approaches surveyed above modifies the splitting criterion itself to detect synergy, and it is that gap which the present study addresses. The aim of this work is to develop and evaluate a decision tree induction algorithm that accounts for feature interactions directly within the splitting criterion, while preserving the univariate, auditable structure that makes single trees useful. This aim distinguishes SCMI-Tree from recent interaction-aware alternatives, which capture feature interactions either by leaving the axis-aligned family altogether, as oblique and soft trees do through weighted combinations of features (Norouzi *et al.*, 2015; Karhikeyan *et al.*, 2022), or by treating interactions only as a pre-processing feature-selection step ahead of an unchanged induction algorithm (Brown *et al.*, 2012; Beraha *et al.*, 2019). In contrast, the present method keeps each split univariate and instead embeds an interaction-information term inside the splitting criterion itself, so that synergy is exploited during induction without sacrificing the readability of the resulting rules. The specific contributions are as follows.

First, a splitting score, termed as SCMI (synergistic conditional mutual information), is proposed. It combines the standard mutual information of a candidate split with reward terms for synergy with ancestor splits and penalty terms for redundancy with them. The penalty is included because, in its absence, the algorithm tends to chase synergy aggressively even on data where redundancy is more common, which sometimes harms accuracy.

Second, a dataset-level synergy score is introduced that estimates the prevalence of feature interactions in the training data. It is computed from a random sample of feature pairs, so its cost is bounded

regardless of dataset size.

Third, an adaptive lookahead mechanism is developed that uses this score to decide whether to engage a single-step forward evaluation, addressing the root XOR problem when it is present and skipping the overhead otherwise.

2. METHODS AND MATERIALS

Background and Notation:

Consider a supervised classification problem with training data $\{(x_i, y_i)\}$ for $i = 1, \dots, n$, where $x_i \in \mathbb{R}^d$ is a feature vector and $y_i \in \{1, \dots, K\}$ is a class label. For random variables X and Y , the Shannon entropy $H(Y) = -\sum p(y) \log_2 p(y)$ measures uncertainty, and the conditional entropy $H(Y | X)$ measures the remaining uncertainty after observing X . The mutual information $I(X; Y) = H(Y) - H(Y | X)$ is the reduction. The interaction information of two features with respect to a target, which is central to what follows, is given in Equation (1). The decomposition of the information that a set of features carries about a target into synergistic and redundant components is formalized by the partial information decomposition of Williams and Beer (2010). The synergy and redundancy terms used below can be read as a pairwise, tree-oriented instance of that broader framework. Throughout, an uppercase X_j denotes the random variable associated with feature j and a lowercase x_j its realized value on a particular sample; a split indicator $S_j = 1[X_j \leq t]$ is itself a binary random variable. The ordered list of ancestor splits on the path from the root to the current node is written P , and a generic ancestor split is indexed by p , with feature index j_p and threshold t_p .

$$II(X_1; X_2; Y) = I(X_1; Y) + I(X_2; Y) - I((X_1, X_2); Y) \quad (1)$$

When the interaction information is positive, the two features share information about Y , so observing both gives less than the sum of observing each separately. This is termed redundancy. When it is negative, the two features jointly carry more information than the sum of their marginal contributions, which is the situation termed synergy. XOR is the extreme case, since each feature is marginally independent of Y , so that $I(X_i; Y) = 0$, while $I((X_1, X_2); Y)$ is large and the interaction information is therefore strongly negative. The symmetric formulation above is used rather than the equivalent asymmetric form $I(X_1; Y) - I(X_1; Y | X_2)$, because in preliminary experiments the symmetric version proved considerably more stable under finite-sample discretization.

All information quantities were estimated by discretizing continuous features into eight quantile bins and using plug-in estimators of the relevant probabilities. Better estimators exist in the literature (Kraskov et al., 2004; Belgazi et al., 2018), but for the kinds of comparisons required inside the tree-induction loop, plug-in estimation is adequate and fast.

The SCMI Splitting Score:

At each internal node, the standard procedure is to enumerate (feature, threshold) candidates and select the one that maximizes an impurity-reduction criterion on the local subset. SCMI-Tree changes the criterion. Let P denote the ordered list of feature, threshold splits along the path from the root to the current node. Let X_j be a candidate feature with threshold t under consideration and let $S_j = 1[X_j \leq t]$ be the resulting binary split indicator. The score is defined in Equation (2), where the synergy and redundancy terms aggregate pairwise interaction information against each ancestor split S_p as given in Equations (3) and (4). Intuitively, the synergy bonus rewards a candidate split whose combination with an ancestor split is more informative about the label than the two splits are separately, which is the signature of an interaction such as XOR; the redundancy penalty discounts a candidate that mostly repeats information an ancestor split has already supplied. The weights λ_{syn} and λ_{red} control how strongly each effect influences the choice of split, and setting both to zero recovers a purely marginal, mutual-information criterion.

$$SCMI(X_j, t | P) = I(S_j; Y) + \lambda_{syn} \cdot Syn(S_j, P) - \lambda_{red} \cdot Red(S_j, P) \quad (2)$$

$$Syn(S_j, P) = (1/|P|) \sum_p \max(0, -II(S_j, S_p; Y)) \quad (3)$$

$$Red(S_j, P) = (1/|P|) \sum_p \max(0, +II(S_j, S_p; Y)) \quad (4)$$

At the root, P is empty, both extra terms vanish, and the score reduces to mutual information, which is the standard information-gain criterion of ID3. At deeper nodes, candidates that interact synergistically with ancestors receive a bonus, while candidates that are redundant with ancestors are penalised. Values of $\lambda_{syn} = 1.0$ and $\lambda_{red} = 0.5$ were used throughout, chosen on a held-out partition of XOR-easy during preliminary tuning. One detail to note is that, to keep per-node cost in check, path sums were computed only with respect to the two most recent ancestors rather than all ancestors, since initial experiments indicated that interactions with distant ancestors are noisy and not worth the computational cost.

The Lookahead Term:

There is one situation that Equation (2) cannot address. Suppose the root split has to select a feature that, on its own, reveals nothing about Y . XOR at the root is exactly this case, since every feature individually has zero mutual information with the target and the SCMI score gives no preference among them. The remedy is a small auxiliary term. For a candidate (X_j, t) , let the left and right child index sets be L and R . The question posed is which of the other features k gives the largest mutual information when measured separately on the left and right children, with each side weighted by its sample share, as given in Equation (5).

$$\text{Lookahead}(X_j, t) = \max_{k \neq j} [w_L \cdot I(X_k; Y | \text{left}) + w_R \cdot I(X_k; Y | \text{right})] \quad (5)$$

Here w_L and w_R are the proportions of samples falling into each child. The combined score, with lookahead engaged, becomes $SCMI(X_j, t | P) + \eta \cdot \text{Lookahead}(X_j, t)$, with η fixed at 0.5. This helps with XOR because an XOR-style root split is itself useless from a marginal point of view, but after the split the data are partitioned so that one of the originally uninformative features becomes informative on at least one child. A marginal score, looking only at the candidate itself, misses this entirely, whereas lookahead detects it.

Dataset Synergy Score and Adaptive Lookahead:

Lookahead should not always be applied. Consider the Breast Cancer Wisconsin data, in which clinical features such as radius, texture and concavity each carry strong marginal information about malignancy, the synergy structure is essentially absent, and lookahead would add overhead without any gain. By contrast, in XOR-hard, each individual feature is nearly uninformative at the root. Consequently, without lookahead, the algorithm selects a split essentially at random, causing the entire tree to be built on a poor foundation. It is therefore desirable for the algorithm to determine which of these two regimes applies before any tree is grown.

The mechanism adopted is a dataset-level synergy score, obtained by sampling feature pairs and measuring synergy strength normalized against each pair's marginal informativeness, as defined in Equation (6). To build intuition for this quantity, consider two extremes. In a dataset whose pairs behave like the exclusive-or problem, each feature on its own is almost uninformative about the label while the pair is fully informative, so the numerator, which captures synergy, is large relative to the marginal terms in the denominator, and $S(D)$ is high. In a dataset whose features are each independently predictive, such as Breast Cancer, the pair adds little beyond what the individual features already provide, the synergistic numerator is close to zero, and $S(D)$ is low. The score can therefore be read as the average amount of label information a dataset hides in feature interactions, expressed as a fraction of what its individual features already reveal.

$$S(D) = (1/|\Pi|) \sum_{(i,j) \in \Pi} [\max(0, -II(X_i, X_j; Y)) / \max(II(X_i; Y), II(X_j; Y), \epsilon)] \quad (6)$$

Here Π is a random sample of up to fifty feature pairs and $\epsilon = 10^{-6}$ prevents division by zero. The quantity $S(D)$ is computed once before any node is split and is not revisited. The decision rule is binary, where lookahead is engaged if $S(D) \geq \tau$ and disabled otherwise. To avoid selecting τ on the same datasets used for evaluation, which would constitute a form of data leakage, the threshold is calibrated on a separate synthetic suite whose generating seeds are disjoint from those of every evaluation dataset. This suite contains marginal-signal problems, whose features are individually informative, together with synergy-rich problems of the XOR and parity type; τ is placed in the gap that separates the two groups, taken as the geometric mean of the largest $S(D)$ among the marginal problems and the smallest $S(D)$ among the synergy problems, so that it reflects the natural break between the two regimes rather than any property of the evaluation data. The calibrated value is $\tau^* = 4.55$, and because the gap between the two regimes is wide, the exact position of the threshold within it does not change which datasets receive lookahead; a sensitivity analysis over a range of τ values (Figure 8) confirms that the reported results are stable throughout the gap. Under this calibration, the evaluation datasets separate cleanly: datasets dominated by marginal signal, such as Breast Cancer ($S = 0.19$), Digits ($S = 0.09$), Ionosphere ($S = 0.43$), the independent-features synthetic dataset ($S = 2.58$), and Sonar ($S = 3.00$), fall on the disabled side, whereas synergy-rich datasets, including Parity-3 ($S = 6.46$), XOR-hard ($S = 12.74$), and XOR-easy ($S = 17.84$), fall on the enabled side.

Algorithmic Procedure:

The training procedure is presented below in a standard algorithmic format. Algorithm 1 describes the complete training workflow; Algorithm 2 defines the node-level SCMI scoring rule; and Algorithm 3 estimates the dataset-level synergy score used to decide whether adaptive lookahead should be activated. In the algorithms, $MI(\cdot)$ denotes mutual information and $II(\cdot)$ denotes interaction information.

Inference is identical to standard tree inference, where the model walks from root to leaf using the stored (feature, threshold) at each internal node and then returns the majority class of the leaf. Training complexity is $O(n d t \kappa)$, where n is the sample count, d the feature count, t the number of candidate thresholds per feature, and κ a constant factor that includes the cost of interaction information computation and, when lookahead is enabled, a per-feature scan over each child. With lookahead disabled, κ is small. With lookahead enabled, κ grows roughly linearly with d , so that SCMI-Tree is meaningfully slower than CART on the largest datasets. This behavior is quantified empirically in Section 3.6: measured training time grows approximately linearly in the sample count n for all methods and the feature count d for SCMI-Lookahead, while remaining almost flat for CART, matching the κ dependence above (Table 4 and Figure 6). The effect of the feature count is severe once lookahead is engaged: at a fixed 2,000 samples, SCMI-Lookahead takes about 80 seconds at 10 features but roughly 5,070 seconds at 80 features, whereas CART stays below 0.1 second throughout. Because the adaptive method disables lookahead on marginal-signal data, its cost there is that of SCMI-Base, a few seconds per fit, which is still far above the few milliseconds CART requires but avoids the steep growth that lookahead incurs on high-dimensional data.

Algorithm 1 SCMI-Tree training with adaptive lookahead

Require: training set $D = \{(x_i, y_i)\}$, max_depth, min_samples_split, min_samples_leaf, synergy threshold τ , lookahead weight η

Ensure: trained decision tree T

```

1:  $S\_D \leftarrow \text{ESTIMATE\_SYNERGY}(D)$  ▷ estimate dataset-level synergy  $S\_D$  once, before growing the tree
2:  $\text{use\_lookahead} \leftarrow (S\_D \geq \tau)$  ▷ enable lookahead only when synergy exceeds the calibrated threshold
3:  $T \leftarrow \text{BUILD\_NODE}(D, \text{depth}=0, \text{path}=\{\}, \text{use\_lookahead})$  ▷ recursively build the root with an empty ancestor path
4: return  $T$ 
5: procedure BUILD_NODE( $D_v$ , depth, path, use_lookahead)
6:   if STOPPING_CRITERIA( $D_v$ , depth) then ▷ stop on depth, purity or min-samples limits
7:     return LEAF(MAJORITY_CLASS( $D_v$ ))
8:   end if
9:   best_feature, best_threshold, best_score  $\leftarrow$  null, null, -inf
10:  relevant_path  $\leftarrow$  LAST_K(path, k=2) ▷ restrict the SCMI interaction terms to the last two ancestors
11:  for each feature  $j$  in  $\{1, \dots, d\}$  do
12:    for each admissible threshold  $t$  of feature  $j$  do
13:      if the split violates min_samples_leaf then
14:        continue
15:      end if
16:      score  $\leftarrow$  SCMI_SCORE( $D_v$ ,  $j$ ,  $t$ , relevant_path) ▷ SCMI score: marginal MI plus synergy bonus minus redundancy penalty
17:      if use_lookahead then
18:        score  $\leftarrow$  score +  $\eta * \text{LOOKAHEAD}(D_v, j, t)$  ▷ if enabled, add the one-step lookahead gain weighted by  $\eta$ 
19:      end if
20:      if score > best_score then
21:        best_feature, best_threshold, best_score  $\leftarrow$   $j, t, \text{score}$ 
22:      end if
23:    end for
24:  end for
25:  if best_score  $\leq 0$  then ▷ no split gives positive score  $\rightarrow$  make a leaf
26:    return LEAF(MAJORITY_CLASS( $D_v$ ))
27:  end if
28:   $DL, DR \leftarrow \text{PARTITION}(D_v, \text{best\_feature}, \text{best\_threshold})$  ▷ partition on the winning (feature, threshold) and recurse
29:  left  $\leftarrow \text{BUILD\_NODE}(DL, \text{depth}+1, \text{path} \cup \{(bf, bt)\}, \text{use\_lookahead})$ 
30:  right  $\leftarrow \text{BUILD\_NODE}(DR, \text{depth}+1, \text{path} \cup \{(bf, bt)\}, \text{use\_lookahead})$ 
31:  return INTERNAL_NODE(best_feature, best_threshold, left, right)
32: end procedure

```

Algorithm 2 Node-level SCMI scoring rule**Require:** node subset D_v , candidate feature j , threshold t , ancestor path P **Ensure:** SCMI score for the candidate split

```

1:  $S_j \leftarrow 1[X_j \leq t]$  for all samples in  $D_v$  ▷ binary split indicator for the candidate feature
2:  $\text{base} \leftarrow MI(S_j; Y)$  ▷ marginal term: MI between the split and the label
3: if  $P$  is empty then
4:   return base ▷ at the root there are no ancestors, so return the marginal term
5: end if
6:  $\text{syn} \leftarrow 0$ ;  $\text{red} \leftarrow 0$ 
7: for each ancestor split  $(j_p, t_p)$  in  $P$  do ▷ accumulate interaction with each ancestor split on the path
8:    $S_p \leftarrow 1[X_{j_p} \leq t_p]$  for all samples in  $D_v$ 
9:    $ii \leftarrow II(S_j, S_p; Y)$  ▷ interaction information between candidate, ancestor and label
10:   $\text{syn} \leftarrow \text{syn} + \max(0, -ii)$  ▷ synergy bonus: reward negative II (joint > sum of parts)
11:   $\text{red} \leftarrow \text{red} + \max(0, +ii)$  ▷ redundancy penalty: discount positive II (overlapping information)
12: end for
13:  $\text{syn} \leftarrow \text{syn}/|P|$ ;  $\text{red} \leftarrow \text{red}/|P|$  ▷ average over the ancestors so path length does not bias the score
14: return  $\text{base} + \lambda_{\text{syn}} * \text{syn} - \lambda_{\text{red}} * \text{red}$  ▷ combine marginal, synergy and redundancy with their weights

```

Algorithm 3 Dataset-level synergy estimation**Require:** training set D , number of sampled feature pairs m , constant ϵ **Ensure:** dataset-level synergy score S_D

```

1:  $\Pi \leftarrow \text{RANDOM\_SAMPLE}(\{(i, j) : 1 \leq i < j \leq d\}, m)$  ▷ sample up to m feature pairs to keep the estimate cheap
2:  $\text{ratios} \leftarrow \text{empty list}$ 
3: for each feature pair  $(i, j)$  in  $\Pi$  do
4:    $mi\_i \leftarrow MI(X_i; Y)$  ▷ marginal informativeness of each feature in the pair
5:    $mi\_j \leftarrow MI(X_j; Y)$ 
6:    $ii \leftarrow II(X_i, X_j; Y)$  ▷ interaction information of the pair with respect to the label
7:    $\text{synergy} \leftarrow \max(0, -ii)$  ▷ keep only the synergistic part (negative II)
8:    $\text{denominator} \leftarrow \max(mi\_i, mi\_j, \epsilon)$  ▷ normalise by the stronger marginal so scores are comparable
9:    $\text{append}(\text{synergy}/\text{denominator} \text{ to } \text{ratios})$ 
10: end for
11: return  $\text{MEAN}(\text{ratios})$  ▷ dataset synergy score = mean normalised synergy over sampled pairs

```

Datasets:

Eight datasets were used, chosen to vary in feature-interaction structure rather than to cover as many real benchmarks as possible. Four are real datasets. Breast Cancer Wisconsin comprises 569 samples with 30 features and 2 classes and represents a medical diagnosis task. Digits comprise 1797 handwritten digit images with 64-pixel features and 10 classes. Sonar comprises 208 samples with 60 acoustic-frequency features and requires classifying metal against rock from sonar returns. Ionosphere comprises 351 samples with 34 features and requires classifying radar returns by atmospheric structure.

The remaining four are synthetic and were used to probe behavior in controlled regimes. Synth-Independent has eight informative independent features and four uninformative ones across three classes with 800 samples and serves as a low-synergy control. Parity-3 has three binary features whose exclusive-or is the label, plus five irrelevant features, with 1200 samples. XOR-easy has two continuous features whose sign-XOR is the label, with four irrelevant features and 5% label noise, across 1500 samples. XOR-hard is identical except that it has eight irrelevant features and 10% noise.

Methods Compared:

Five methods were compared, all configured with a maximum depth of 5, a minimum of 10 samples per split and a minimum of 5 samples per leaf. CART-Gini is the standard scikit-learn implementation with Gini impurity. ID3-Entropy is the same library using the entropy criterion. SCMI-Base is the proposed score without lookahead. SCMI-Lookahead is the proposed score with lookahead always enabled. SCMI-Adaptive is the proposed score with lookahead engaged according to the dataset synergy score. Hyperparameters were

held identical across all methods to ensure that any differences arise from the splitting criterion alone.

Evaluation Protocol:

Stratified five-fold cross-validation was used, with the same random seed applied to all methods on each fold. Per-fold accuracies were aggregated into a mean and standard deviation. For significance testing, paired t-tests were run at three thresholds, namely $p < 0.10$, $p < 0.05$ and $p < 0.01$. Effect sizes are reported as Cohen's d , with the conventional value of $d \geq 0.8$ marking the large-effect boundary. Per-dataset ranks were obtained by sorting mean accuracies in descending order, and aggregating across the eight datasets gives each method an average rank. All experiments were implemented in Python using the scikit-learn library.

3. RESULTS**Accuracy Across Datasets:**

Accuracy results for all five methods across the full dataset suite are presented in Figure 2, with the estimated synergy score S labelled above each dataset's bar cluster. Two groupings emerge. For Breast Cancer, Digits, Sonar, Ionosphere and Synth-Independent, all of which have low or moderate synergy scores, the five methods perform within a few percentage points of each other and no method dominates. For Parity-3, XOR-easy and XOR-hard, the gap widens. The two SCMI variants with lookahead reach 95% and 89% on the XOR tasks, while CART and ID3 remain between 53% and 69%. The per-dataset numbers are reported in Table 1, presenting the mean and standard deviation of every method together with the synergy score S , the significance markers and the effect sizes.

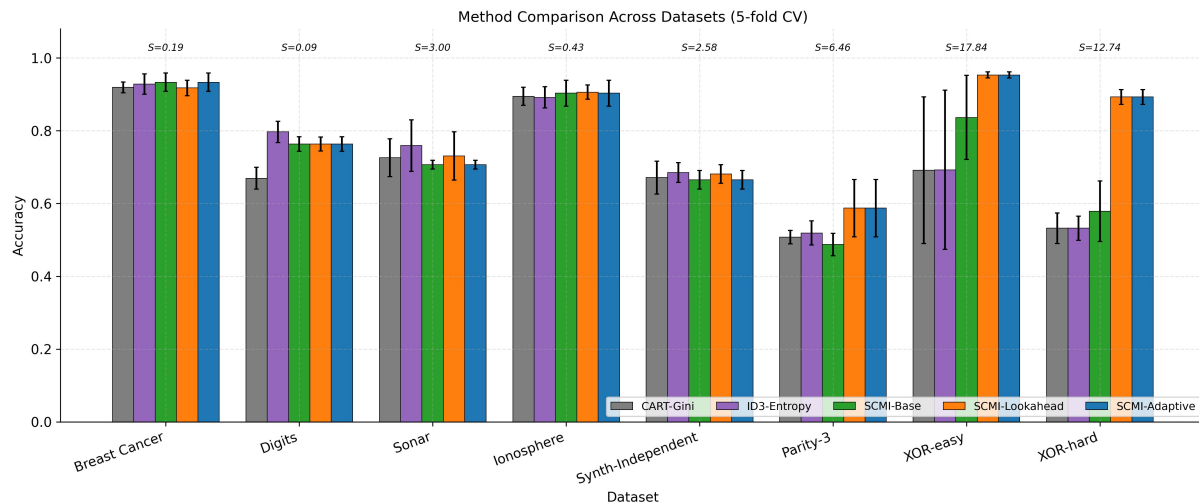


Figure 2: Five-fold cross-validated mean accuracy, with standard deviation error bars, for all five tree variants plotted per dataset. The synergy score S is annotated above each bar group.

Table 1: Per-dataset classification accuracy of SCMI-Tree and baseline splitting criteria, reported as mean $\pm$ standard deviation under stratified five-fold cross-validation.

Dataset	S	CART-Gini	ID3-Entropy	SCMI-Base	SCMI-Lookahead	SCMI-Adaptive	Cohen's d
Breast Cancer	0.19	0.919 $\pm$ 0.015	0.928 $\pm$ 0.028	0.933 $\pm$ 0.025	0.917 $\pm$ 0.021	0.933 $\pm$ 0.025	-0.08
Digits	0.09	0.669 $\pm$ 0.030	0.797 $\pm$ 0.029	0.763 $\pm$ 0.020	0.763 $\pm$ 0.019***	0.763 $\pm$ 0.020	2.80
Sonar	3.00	0.726 $\pm$ 0.052	0.759 $\pm$ 0.071	0.707 $\pm$ 0.012	0.731 $\pm$ 0.066	0.707 $\pm$ 0.012	0.10
Ionosphere	0.43	0.895 $\pm$ 0.025	0.892 $\pm$ 0.029	0.903 $\pm$ 0.036	0.906 $\pm$ 0.020	0.903 $\pm$ 0.036	0.49
Synth-Independent	2.58	0.671 $\pm$ 0.045	0.685 $\pm$ 0.027	0.665 $\pm$ 0.026	0.681 $\pm$ 0.025	0.665 $\pm$ 0.026	0.34
Parity-3	6.46	0.508 $\pm$ 0.019	0.519 $\pm$ 0.033	0.487 $\pm$ 0.031	0.588 $\pm$ 0.079*	0.588 $\pm$ 0.079	0.96
XOR-easy	17.84	0.691 $\pm$ 0.201	0.693 $\pm$ 0.219	0.837 $\pm$ 0.116	0.953 $\pm$ 0.008*	0.953 $\pm$ 0.008	1.20
XOR-hard	12.74	0.532 $\pm$ 0.042	0.532 $\pm$ 0.034	0.579 $\pm$ 0.083	0.893 $\pm$ 0.020***	0.893 $\pm$ 0.020	5.79

Bold indicates the highest mean accuracy per dataset. Significance markers denote paired t -tests of SCMI-Lookahead against CART-Gini, where *** indicates $p < 0.01$ and * indicates $p < 0.10$. Cohen's d is reported for SCMI-Lookahead against CART-Gini. The SCMI-Adaptive column reports the adaptive variant, which by construction equals SCMI-Lookahead on every dataset with $S \geq \tau^*$ and SCMI-Base otherwise (Section 2.4).

SCMI-Lookahead achieved statistically significant gains over CART-Gini on four of the eight datasets, namely Digits at $p < 0.01$, Parity-3 and XOR-easy at $p < 0.10$, and XOR-hard at $p < 0.01$. The XOR-hard result has a Cohen's d of 5.79, reflecting that every fold shows the same large gap with very little variance. ID3-Entropy also exceeded CART-Gini on several datasets, most notably Digits at 79.7%. The SCMI-Base variant recorded gains over CART-Gini on Breast Cancer at 93.3% against 91.9% and on XOR-easy at 83.7% but fell below the full lookahead variant on the most synergy-rich datasets.

Validating the Synergy Scoretitle:

Figure 3 plots each of the eight datasets as a single point, with the horizontal coordinate equal to its estimated $S(D)$, on a logarithmic scale, and the vertical coordinate equal to the accuracy advantage of SCMI-Adaptive over CART-Gini on that dataset. Under the calibrated threshold, the four datasets with the lowest synergy scores, Breast Cancer, Ionosphere, the independent-features synthetic set and Sonar, cluster around the zero line, since on these datasets the adaptive switch disables lookahead and SCMI-Adaptive tracks CART to within about two accuracy points. The three synergy-rich datasets, Parity-3, XOR-easy and XOR-hard, show the large positive advantages expected when lookahead is engaged. Digits is the one low-synergy dataset with a clear positive advantage, which reflects the base SCMI criterion already outperforming Gini on that dataset rather than any synergy effect.

Table 2: Cross-validation stability of the dataset synergy score $S(D)$. The full-sample value is shown alongside the mean and standard deviation of $S(D)$ recomputed within each of the five folds, and the number of folds on which adaptive lookahead engaged.

Dataset	S (full sample)	S (per-fold mean $\pm$ SD)	Folds engaged
Breast Cancer	0.193	0.234 $\pm$ 0.017	0 / 5
Digits	0.092	0.110 $\pm$ 0.002	0 / 5
Sonar	2.997	3.333 $\pm$ 0.187	0 / 5
Ionosphere	0.432	0.487 $\pm$ 0.071	0 / 5
Synth-Independent	2.579	2.861 $\pm$ 0.227	0 / 5
Parity-3	6.463	6.558 $\pm$ 1.008	5 / 5
XOR-easy	17.840	18.887 $\pm$ 4.406	5 / 5
XOR-hard	12.739	10.801 $\pm$ 0.601	5 / 5

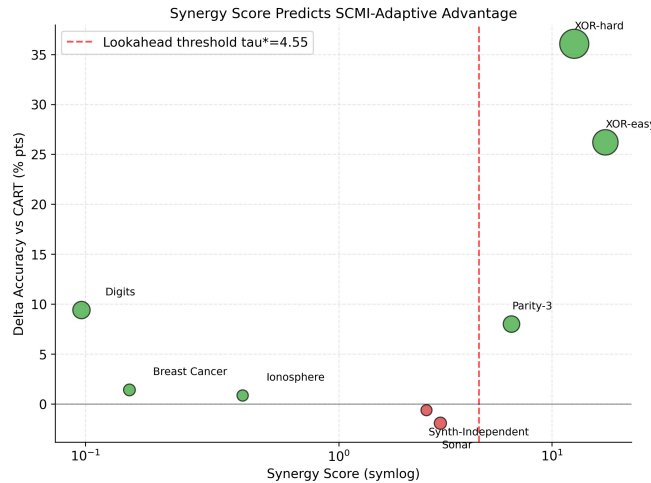


Figure 3: Estimated dataset synergy score plotted against the empirical accuracy advantage of SCMI-Adaptive over CART-Gini. The dashed line marks the calibrated lookahead engagement threshold $S = \tau^* = 4.55$.

Method Ranking:

Figure 4 presents the per-dataset rankings aggregated across the suite. SCMI-Lookahead ranked best overall at 1.88, followed by ID3-Entropy at 2.75, SCMI-Adaptive at 3.12, SCMI-Base at 3.12 and CART-Gini at 4.12. The wide error bar for SCMI-Lookahead reflects its variation across datasets, ranking near the top on synergy-rich data and mid-pack on synergy-poor data. The narrow error bar for CART-Gini reflects that it remained near the bottom throughout.

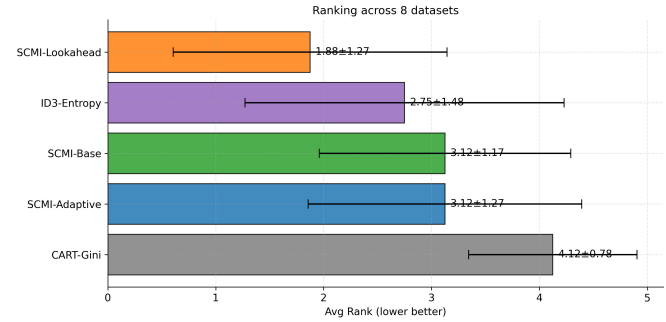


Figure 4: Average rank of each method across the eight datasets, where lower is better. Error bars show the standard deviation of the rank.

Ablation Studies:

Four ablation studies were conducted on the XOR task, and all four are presented in Figure 5.

Panel (a) reports accuracy as a function of label noise. Without label noise, SCMI attained 99.7% accuracy while CART and ID3 remained in the sixties. As noise increased, SCMI degraded gradually but retained its advantage up to 30% noise, beyond which all methods approached chance level.

Panel (b) reports accuracy as a function of the number of irrelevant features. With no noise features, all three methods achieved accuracy ranging from 87% to 89%. As irrelevant features were added, CART and ID3 degraded severely, reaching 53% at eight irrelevant features and remaining at that level for sixteen and thirty-two. SCMI-Adaptive was barely affected, with remaining accuracy between 88% and 90% across all values.

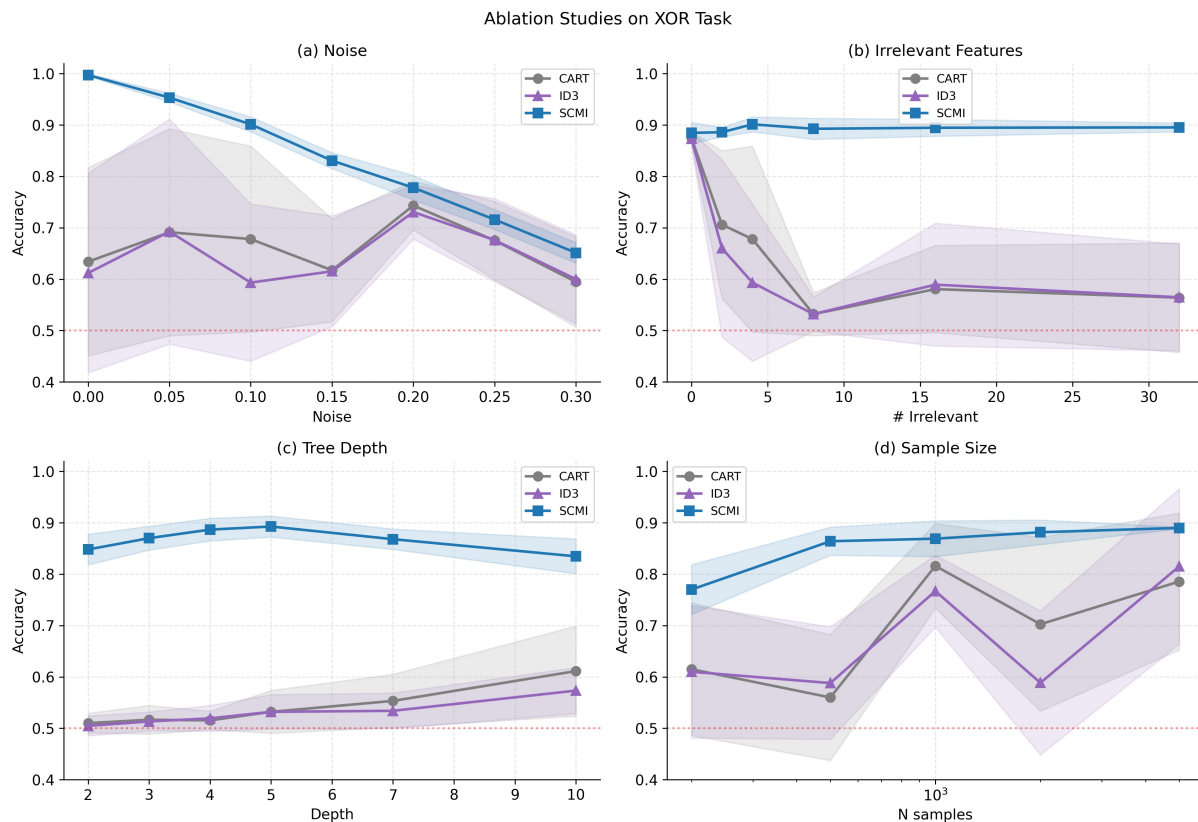


Figure 5: Ablation studies on the XOR task. (a) Effect of label noise rate. (b) Effect of irrelevant features. (c) Effect of tree depth. (d) Effect of training sample size.

Panel (c) reports accuracy as a function of maximum tree depth. SCMI-Adaptive reached 85% at depth 2, while CART and ID3 remained at chance level for depths 2 and 3. The performance of CART and ID3 improved as depth increased, but only moderately, with CART achieving approximately 61% at depth 10, whereas SCMI peaked at around depth 5 and then declined slightly.

Panel (d) reports accuracy as a function of the number of training samples, from 200 to 5000. With only 200 training examples on noisy XOR with four noise features, SCMI achieved 77% while CART and ID3 achieved 62% and 61% respectively. Increasing the sample count improved CART and ID3, with ID3 reaching 82% and CART 79% at 5000 samples, but SCMI maintained an advantage that was largest at small sample sizes.

Tree Complexity:

Accuracy alone does not capture the cost of a model in terms of interpretability, so the size of the induced trees was also measured. Table 3 reports, for each dataset, the mean number of nodes of each method, averaged over the cross-validation folds. All methods were grown under the same depth limit, resulting in trees with a fixed depth of five. Consequently, the comparison focuses on node counts at equal depth, while the number of distinct features used at split points remained comparable across methods. On the marginal-signal datasets, where the adaptive method disables lookahead and therefore reduces to SCMI-Base, the trees are within a few nodes of the CART and ID3 trees (for example, twenty-six against twenty-three nodes on Breast Cancer and twenty-one against twenty-one on Ionosphere), confirming that the synergy-aware criterion does not inflate model complexity when interactions are absent. On the synergy-rich datasets, the picture differs: SCMI-Lookahead does not produce smaller trees than CART and, in fact, uses somewhat more nodes. For example, it uses 61 nodes against 36 on XOR-hard, 55 against 37 on XOR-easy, and 57 against 38 on Parity-3. The larger tree is not a sign of a more complex explanation but of a more accurate one: on XOR-hard the CART tree is small only because it stops at an accuracy near chance, whereas the SCMI-Lookahead tree spends its additional nodes turning the recognised interaction into a correct decision boundary, raising accuracy from 53% to 89% at the same depth. The additional nodes therefore reflect finer partitioning of the same informative features rather than reliance on more of them. The per-method node counts are given in Table 3.

Runtime, Scalability and Sensitivity:

Beyond accuracy, three further properties of the method were measured: its training-time cost, how that cost scales with the size of the problem, and its sensitivity to the hyperparameters. These experiments were added in response to the reviewers and are reported here.

Training time: Table 4 reports the mean per-fold fit time on each benchmark. CART and ID3 train in a few milliseconds. SCMI-Base, which computes interaction information at every candidate split, takes a few seconds, and SCMI-Lookahead, which additionally scans all remaining features on both children of every candidate, takes from about

twenty seconds to about eighteen minutes. The adaptive method incurs the lookahead cost only on the synergy-rich datasets and otherwise runs at the speed of SCMI-Base.

Scalability: Figure 6 and Table 5 report how training time grows as the sample count and the feature count are varied independently on a synthetic XOR task. Training time grows roughly linearly in the number of samples for all methods. As the number of features increases, runtime grows only modestly for CART but rises steeply for SCMI-Lookahead. At 2,000 samples, the runtime of SCMI-Lookahead increases from approximately 80 seconds with 10 features to roughly 5,000 seconds with 80 features. This behavior provides empirical evidence for the per-feature complexity factor discussed in Section 2.5 and explains why the adaptive method restricts lookahead to datasets for which it is warranted.

Hyperparameter sensitivity: Figure 7 reports how accuracy on one synergy-rich and one marginal dataset varies as each hyperparameter is changed while the others are held at their defaults. Accuracy is completely insensitive to the synergy weight and the redundancy weight over the tested range, so these two parameters require no tuning. The lookahead weight has a clear optimum near 0.25 to 0.50, the region containing the default of 0.5, with accuracy falling off if the weight is set substantially higher.

Threshold sensitivity: Figure 8 shows the accuracy of the adaptive method as the engagement threshold is swept across the interval separating the marginal and synergy-rich regimes. Because the two regimes are separated by a wide margin, every threshold in that interval produces the same regime assignment and hence the same accuracy, confirming that the exact value of the calibrated threshold within the gap does not matter.

4. DISCUSSION

When SCMI Helps:

A clear trend emerges across the eight datasets. When feature interactions are present in the underlying problem, SCMI outperforms the baseline methods; when they are absent, SCMI matches their performance. The precomputed dataset synergy score $S(D)$ correctly anticipates these two regimes. Datasets with marginally informative features, namely Breast Cancer at $S = 0.19$, Digits at 0.09 and Ionosphere at 0.43, fall in the region where SCMI neither wins nor loses. Datasets with genuine interaction structure, namely Parity-3 at 6.46, XOR-easy at 17.84 and XOR-hard at 12.74, are where the gap between SCMI and the baselines opens.

The overall picture is nonetheless mixed, and there are three datasets on which SCMI gains nothing. On Breast Cancer it finishes tied with CART and ID3, with all five methods bunched within about 1.6 percentage points. On Sonar, ID3-Entropy comes out marginally ahead and SCMI sits in the middle of the pack. On Synth-Independent, which has eight informative features and no synergy by design, the methods are again indistinguishable. The common thread is low or moderate dataset synergy, which means that the lookahead module either does not switch on at all, or switches on but finds no interaction signal worth exploiting. This is precisely what the design intends, namely a fall-back to standard performance when synergy is absent, but it does establish that SCMI does not dominate everywhere.

Table 3: Mean number of tree nodes for each method, averaged over the five cross-validation folds. All trees were grown to the same maximum depth of five, and the number of distinct features used at split points was comparable across methods (typically within one or two features), so node count is the informative dimension of complexity here.

Dataset	CART-Gini	ID3-Entropy	SCMI-Base	SCMI-Lookahead
Breast Cancer	22.6	23.0	26.2	29.0
Digits	49.8	55.0	57.0	58.2
Sonar	24.6	21.0	24.2	27.0
Ionosphere	21.0	19.4	20.6	26.6
Synth-Independent	51.8	53.4	53.8	60.2
Parity-3	37.8	27.0	50.2	57.4
XOR-easy	37.0	35.0	50.6	55.0
XOR-hard	35.8	29.8	55.0	61.4

Table 4: Mean per-fold training time in seconds for each method. SCMI-Adaptive incurs the lookahead cost only on the synergy-rich datasets, where it equals SCMI-Lookahead; on the marginal datasets it disables lookahead and equals SCMI-Base.

Dataset	CART-Gini	ID3-Entropy	SCMI-Base	SCMI-Lookahead	SCMI-Adaptive
Breast Cancer	0.009	0.010	4.43	232.6	4.43
Digits	0.014	0.018	13.20	1084.5	13.20
Sonar	0.005	0.006	6.71	646.1	6.71
Ionosphere	0.005	0.007	3.04	188.8	3.04
Synth-Independent	0.006	0.009	4.05	88.17	4.05
Parity-3	0.005	0.008	3.26	38.82	38.82
XOR-easy	0.006	0.009	2.01	22.15	22.15
XOR-hard	0.008	0.012	3.57	64.05	64.05

Table 5: Single-fit training time in seconds on a synthetic XOR task as the sample count and the feature count are varied independently. The steep growth of SCMI-Lookahead with the feature count is the empirical counterpart of the per-feature factor in the complexity analysis.

Setting	CART-Gini	SCMI-Base	SCMI-Lookahead
<i>Varying sample count n (feature count $d = 20$)</i>			
$n = 500$	0.011	4.47	188.5
$n = 1000$	0.011	6.58	236.2
$n = 2000$	0.023	7.52	302.8
$n = 4000$	0.047	8.93	375.0
$n = 8000$	0.126	11.03	426.9
<i>Varying feature count d (sample count $n = 2000$)</i>			
$d = 10$	0.012	3.70	79.57
$d = 20$	0.022	8.90	324.8
$d = 40$	0.043	14.32	1287.7
$d = 80$	0.089	32.67	5068.0

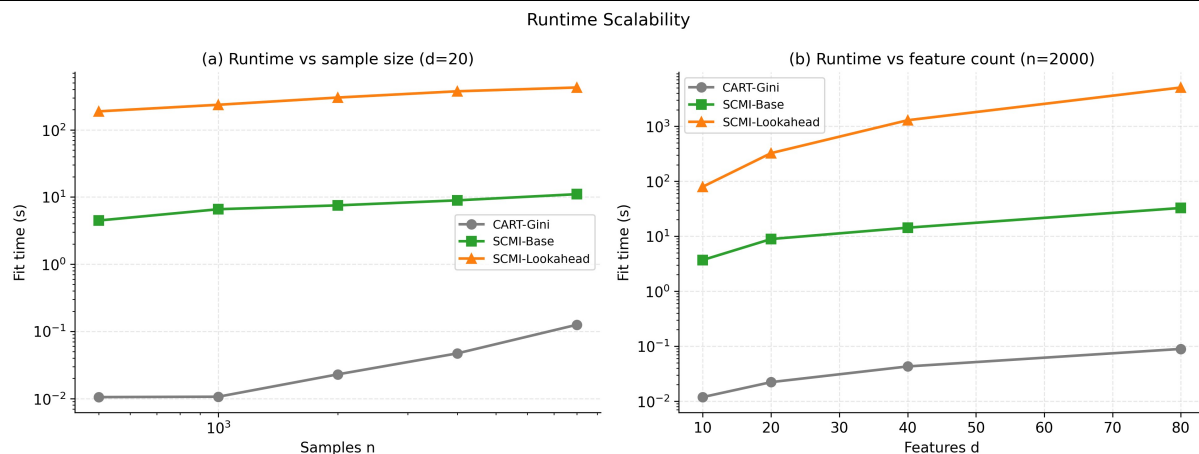


Figure 6: Training time as a function of the number of samples (left) and the number of features (right) on a synthetic XOR task, for CART, SCMI-Base and SCMI-Lookahead. Note the steep growth of the lookahead variant with the feature count.

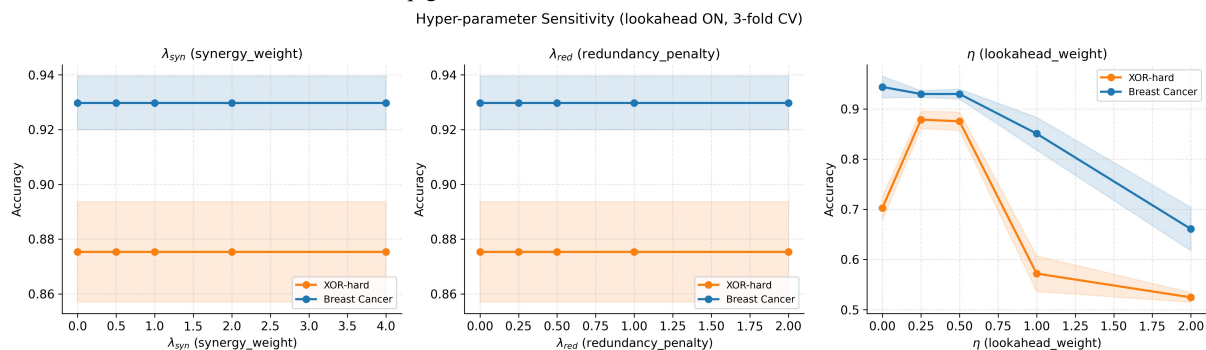


Figure 7: Accuracy on XOR-hard and Breast Cancer as the synergy weight, the redundancy weight and the lookahead weight are each varied about their default values (lookahead enabled, three-fold cross-validation). Accuracy is flat in the first two weights and has a clear optimum in the third.

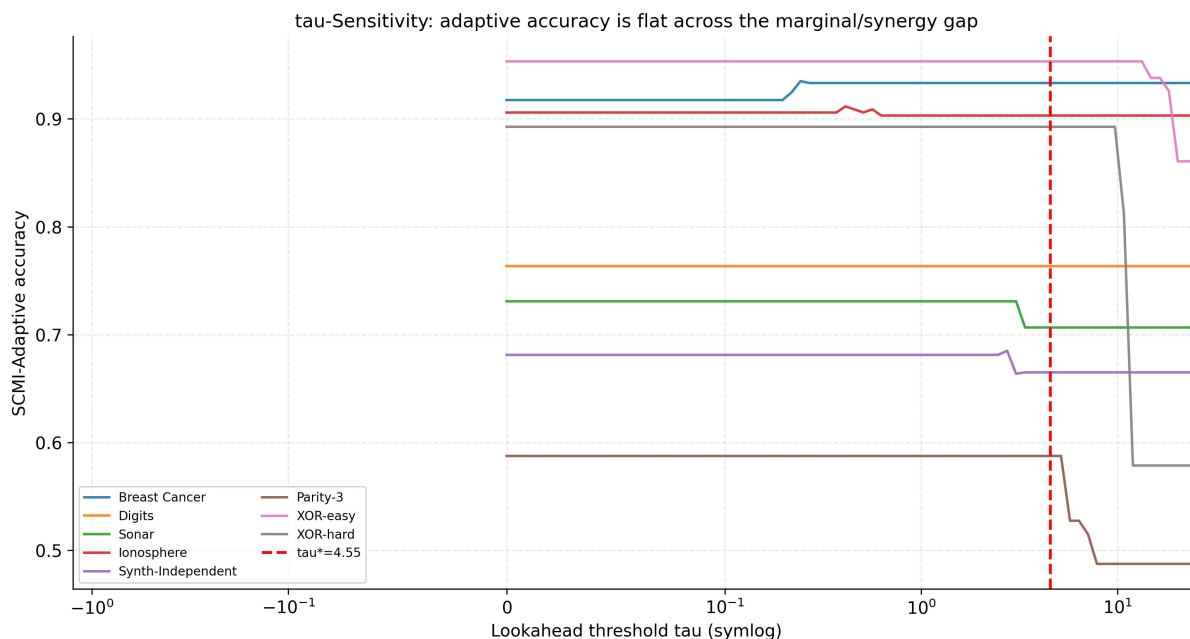


Figure 8: Accuracy on XOR-hard and Breast Cancer as the synergy weight, the redundancy weight and the lookahead weight are each varied about their default values (lookahead enabled, three-fold cross-validation). Accuracy is flat in the first two weights and has a clear optimum in the third.

The exception noted for Sonar in Section 3.2 merits comment. Its synergy score of 3.0 places it on the marginal side of the calibrated threshold, so the adaptive method disables lookahead and reduces to SCMI-Base, which on this dataset finishes slightly below CART and ID3 rather than ahead of them. Given its small size of only 208 samples, there may be insufficient data for the interaction-aware criterion to gain traction, since estimating interaction information reliably requires more observations than estimating marginal information.

This characterization offers practical guidance. When there is reason to expect feature interactions in the data, SCMI is a suitable choice, whereas when the features are believed to contribute largely independently, the simpler standard methods are adequate. The dataset-level synergy score, computed before training, provides an objective check on which of these two regimes applies.

The Role of Adaptive Lookahead:

By construction, SCMI-Adaptive reduces to SCMI-Lookahead on every dataset placed on the synergy side of the calibrated threshold and to SCMI-Base on every dataset placed on the marginal side, so its per-dataset accuracy equals that of the selected variant. This agreement is recorded in the SCMI-Adaptive column of Table 1. The adaptive switch therefore performs as intended, engaging lookahead when the dataset-level synergy score calls for it and leaving it disabled otherwise. This also means, however, that in terms of accuracy alone SCMI-Adaptive offers no separate advantage over running lookahead at all times. The benefit of the adaptive variant is computational rather than statistical, and where training time is not a concern a practitioner may use SCMI-Lookahead directly.

Comparison with Entropy-Based Induction:

ID3-Entropy performed considerably better as a baseline than anticipated. On Digits, SCMI exceeded CART-Gini by approximately twelve percentage points, a difference far too large to be attributed to random variation and consistent with previous findings that entropy-based criteria can outperform Gini impurity on image-like datasets (Raileanu & Stoffel, 2004). This complicates the interpretation somewhat, since comparing SCMI against CART alone would invite the objection that any gain is really entropy outperforming Gini. The rel-

evant observation is that SCMI-Lookahead still ranks better on average than ID3-Entropy, at 1.88 against 2.75 across the eight datasets, so the improvement does not arise from the impurity measure alone and the synergy and lookahead components contribute materially.

Robustness Properties:

The ablation results in Section 3.4 identify two conventional weaknesses of greedy induction that the proposed method largely avoids. The robustness to irrelevant features shown in panel (b) arises because the detection of synergy between two informative features succeeds even in the presence of many noisy distractors, whereas a greedy split based on the marginal distribution of each variable has an increasing chance of being misled as distractors accumulate. The robustness to shallow depth shown in panel (c) is of practical consequence, since for problems with an underlying interaction structure an SCMI tree of depth 2 outperforms a CART tree of depth 10 by roughly 25 percentage points while remaining far easier to comprehend. The slight decline beyond depth 5 is a familiar overfitting effect, in which deeper trees begin to fit noise. The sample-efficiency result in panel (d) is expected, since detecting synergy is the appropriate strategy when only two features of a larger set carry information, and it should require fewer samples than relying on a greedy strategy to encounter the relevant features by chance.

Limitations of the Study:

Several limitations should be acknowledged. The most visible is computational cost. Across the eight benchmark datasets, the measured training time of the deployed adaptive method ranges from approximately 3 to 64 seconds, whereas the full lookahead variant requires between 22 seconds and 18 minutes on the largest dataset. By comparison, CART requires less than 0.02 seconds on the same data (Table 4). Furthermore, the scaling experiments in Section 3.6 show that this computational gap widens as the number of features increases once lookahead is engaged. Two factors drive this behavior. First, interaction information computations require the construction of joint histograms over three discretized variables. Second, when lookahead is enabled, each candidate split triggers a scan across all remaining features in both child nodes. For mid-sized data, this is acceptable, but

for very large datasets it is currently prohibitive without further engineering. Three avenues for improvement are to cache the marginal mutual information of each feature, since it does not change across nodes, to use histogram-style approximations rather than full quantile-bin recomputation, and to prune candidate features whose marginal mutual information is far below the current best.

A second limitation concerns statistical power. Five-fold cross-validation provides a limited basis for significance claims, since each pairwise comparison has effectively five paired observations, which is barely sufficient for a t-test to detect anything but the largest effects. Several of the directional results in Table 1, specifically those significant only at the 0.10 level, might well clear the 0.05 threshold under repeated stratified cross-validation or with a larger benchmark suite, neither of which has yet been undertaken.

A further consideration is the sensitivity of the method to its hyperparameters. The synergy weight λ_{syn} , the redundancy weight λ_{red} , the lookahead weight η and the engagement threshold τ were not tuned per dataset; they were held at fixed default values throughout the main experiments. To check that the reported results do not depend on a fortunate choice of these values, each was varied over a range while the others were held at their defaults, on one synergy-rich and one marginal dataset (Figure 7). The outcome separates the three weights clearly. Accuracy is completely insensitive to the synergy weight and the redundancy weight: over the entire range tested, from zero up to four times the default, the accuracy on both datasets does not change at all, indicating that on these problems the split chosen at each node is already determined by the marginal and lookahead terms and is not altered by the synergy and redundancy adjustments. The lookahead weight, by contrast, has a clear effect with a well-defined optimum: accuracy on XOR-hard rises from 0.70 with the lookahead term switched off to about 0.88 at a weight of 0.25 to 0.50, then falls to 0.52 as the weight is increased to two, and the marginal dataset shows the same decline at large weights. The default value of 0.5 therefore sits at the top of this curve, and the practical guidance is that the lookahead weight should be kept moderate while the synergy and redundancy weights require no tuning. The engagement threshold τ was varied across the interval separating the marginal and synergy-rich regimes (Figure 8). Because this interval is wide, all threshold values within it produce the same regime assignment and, consequently, the same adaptive accuracy. This indicates that the method is largely insensitive to the precise choice of threshold within the gap. Full sensitivity curves are reported in Section 3.6.

A third limitation is the reliance on synthetic data for the largest observed effects. The strongest results occur on XOR-style synthetic tasks. XOR is a legitimate stress test for synergy-aware methods, since a CART accuracy near chance represents a real algorithmic failure rather than a contrived one, but the frequency with which XOR-like patterns occur in real data is an empirical question that remains unresolved here. Identifying real-world datasets with this structure, particularly in biomedical and sensor-fusion domains, is the most important next step.

A fourth limitation is that only single trees were considered. The work is deliberately framed as a single-tree contribution for reasons of interpretability, and no comparison was made against random forests or gradient boosting, which would likely outperform SCMI-Tree on most real datasets. Whether a random forest composed of SCMI trees would outperform a standard CART-based forest on synergy-rich datasets remains an open question and has not yet been investigated.

5. CONCLUSION

This study introduced SCMI-Tree, a decision tree induction algorithm whose splitting score accounts for synergy and redundancy with ancestor splits, and which decides adaptively whether to engage a lookahead step based on a cheaply computed dataset-level synergy score. Across eight datasets, the lookahead variant achieved the best average rank among the methods compared. The gains are particularly large on data with strong feature interactions, where standard greedy trees fail badly. Ablation studies further show that SCMI is robust to large numbers of irrelevant features and to shallow tree depths, which are two known weak points of standard greedy induction.

The broader implication is that the standard decision tree, by

far the most widely deployed interpretable model in tabular machine learning, can be improved meaningfully without sacrificing the properties that make it useful. Treating each candidate split as part of a joint information structure on the path, rather than as an isolated marginal feature, recovers patterns that standard induction does not detect. The splitting score and the synergy detection step proposed here are offered both as a tool for practitioners and as a starting point for further work on interaction-aware tree induction.

Acknowledgment:

The authors thank colleagues at the College of Science, University of Zakho, and Duhok Polytechnic University for helpful discussions on earlier drafts of this work.

Ethical Statment:

This study did not involve human participants, human-derived material, or animal subjects. All datasets analyzed are publicly available, fully anonymized benchmark datasets, or were generated synthetically by the authors. Accordingly, no ethical approval was required from any Human Ethics Committee or any Animal Ethics Committee.

Conflict of Interests:

The authors declare that they have no conflicts of interest related to the publication or funding of this manuscript.

Funding:

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors. No financial support was received for the conduct of this study or the preparation of this manuscript.

Author Contributions:

All authors have reviewed the final version to be published and agreed to be accountable for all aspects of the work. The individual contributions are as follows: T.A.A. and O.M.A.: concept and design; H.J.M. and D.H.S.: acquisition, analysis and interpretation of data; T.A.A., D.H.S. and O.M.A.: manuscript drafting.

6. REFERENCES

- Akash, P. S., Kadir, M. E., Ali, A. A., & Shoyaib, M. (2019). Inter-node Hellinger distance based decision tree. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI-19)* (pp. 1967–1973). DOI:<https://doi.org/10.24963/ijcai.2019/272>
- Anastassiou, D. (2007). Computational analysis of the synergy among multiple interacting genes. *Molecular Systems Biology*, 3(1), 83. DOI:<https://doi.org/10.1038/msb4100124>
- Belghazi, M. I., Baratin, A., Rajeshwar, S., Ozair, S., Bengio, Y., Courville, A., & Hjelm, R. D. (2018). Mutual information neural estimation. In *Proceedings of the 35th International Conference on Machine Learning* (Vol. 80, pp. 531–540). DOI:<https://doi.org/10.48550/arXiv.1801.04062>
- Beraha, M., Metelli, A. M., Papini, M., Tirinzoni, A., & Restelli, M. (2019). Feature selection via mutual information: New theoretical insights. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN)* (pp. 1–9). DOI:<https://doi.org/10.1109/IJCNN.2019.8852410>
- Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5–32. DOI:<https://doi.org/10.1023/A:1010933404324>

- Breiman, L., Friedman, J., Olshen, R. A., & Stone, C. J. (1984). Classification and regression trees. Brooks/Cole, Wadsworth, Monterey, CA, 358 pp. DOI:<https://doi.org/10.1201/9781315139470>
- Brown, G., Pocock, A., Zhao, M.-J., & Luján, M. (2012). Conditional likelihood maximisation: A unifying framework for information theoretic feature selection. *Journal of Machine Learning Research*, 13(1), 27–66.
- Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785–794). DOI:<https://doi.org/10.1145/2939672.2939785>
- Cieslak, D. A., Hoens, T. R., Chawla, N. V., & Kegelmeyer, W. P. (2012). Hellinger distance decision trees are robust and skew-insensitive. *Data Mining and Knowledge Discovery*, 24(1), 136–158. DOI:<https://doi.org/10.1007/s10618-011-0222-1>
- Elomaa, T., & Malinen, T. (2003). On lookahead heuristics in decision tree learning. In *Foundations of Intelligent Systems (ISMIS 2003), Lecture Notes in Computer Science* (Vol. 2871, pp. 445–453). Springer, Berlin. DOI:https://doi.org/10.1007/978-3-540-39592-8_63
- Esmeir, S., & Markovitch, S. (2007). Anytime learning of decision trees. *Journal of Machine Learning Research*, 8(5), 891–933.
- Grinsztajn, L., Oyallon, E., & Varoquaux, G. (2022). Why do tree-based models still outperform deep learning on typical tabular data? In *Advances in Neural Information Processing Systems* (Vol. 35, pp. 507–520). DOI:<https://doi.org/10.48550/arXiv.2207.08815>
- Karthikeyan, A., Jain, N., Natarajan, N., & Jain, P. (2022). Learning accurate decision trees with bandit feedback via quantized gradient descent. *Transactions on Machine Learning Research*, 2022(6), 1–28. DOI:<https://doi.org/10.48550/arXiv.2102.07567>
- Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T.-Y. (2017). LightGBM: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems* (Vol. 30, pp. 3146–3154).
- Kraskov, A., Stögbauer, H., & Grassberger, P. (2004). Estimating mutual information. *Physical Review E*, 69(6), 066138. DOI:<https://doi.org/10.1103/PhysRevE.69.066138>
- Lyon, R. J., Brooke, J. M., Knowles, J. D., & Stappers, B. W. (2014). Hellinger distance trees for imbalanced streams. In *Proceedings of the 22nd International Conference on Pattern Recognition (ICPR)* (pp. 1969–1974). DOI:<https://doi.org/10.1109/ICPR.2014.344>
- McGill, W. J. (1954). Multivariate information transmission. *Psychometrika*, 19(2), 97–116. DOI:<https://doi.org/10.1007/BF02289159>
- Murthy, S. K., Kasif, S., & Salzberg, S. (1994). A system for induction of oblique decision trees. *Journal of Artificial Intelligence Research*, 2(1), 1–32. DOI:<https://doi.org/10.1613/jair.63>
- Norouzi, M., Collins, M., Johnson, M. A., Fleet, D. J., & Kohli, P. (2015). Efficient non-greedy optimization of decision trees. In *Advances in Neural Information Processing Systems* (Vol. 28, pp. 1729–1737).
- Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, 1(1), 81–106. DOI:<https://doi.org/10.1007/BF00116251>
- Quinlan, J. R. (1993). C4.5: Programs for machine learning. Morgan Kaufmann, San Francisco, CA, 302 pp. DOI:<https://doi.org/10.1016/C2009-0-27846-9>
- Raileanu, L. E., & Stoffel, K. (2004). Theoretical comparison between the Gini index and information gain criteria. *Annals of Mathematics and Artificial Intelligence*, 41(1), 77–93. DOI:<https://doi.org/10.1023/B:AMAI.0000018580.96245.c6>
- Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5), 206–215. DOI:<https://doi.org/10.1038/s42256-019-0048-x>
- Tan, H., Wang, G., Wang, W., & Zhang, Z. (2022). Feature selection based on distance correlation: A filter algorithm. *Journal of Applied Statistics*, 49(2), 411–426. DOI:<https://doi.org/10.1080/02664763.2020.1815672>
- Tax, T. M. S., Mediano, P. A. M., & Shanahan, M. (2017). The partial information decomposition of generative neural network models. *Entropy*, 19(9), 474. DOI:<https://doi.org/10.3390/e19090474>
- Vergara, J. R., & Estévez, P. A. (2014). A review of feature selection methods based on mutual information. *Neural Computing and Applications*, 24(1), 175–186. DOI:<https://doi.org/10.1007/s00521-013-1368-0>
- Wickramarachchi, D. C., Robertson, B. L., Reale, M., Price, C. J., & Brown, J. (2016). HHCART: An oblique decision tree. *Computational Statistics and Data Analysis*, 96(1), 12–23. DOI:<https://doi.org/10.1016/j.csda.2015.11.006>