Can Interpretable Clustering Trees Be Certified? Provable Bounds under Discretized Splits

Anonymous authors
Paper under double-blind review

Abstract

Interpretable clustering trees provide explicit rules for cluster membership, but existing solvers typically rely on heuristic search, leaving the quality of the returned tree unknown within the prescribed tree class. We present **C**ertified **I**nterpretable **C**lustering **T**rees (CICT), a branch-and-bound method that maximizes a regularized Dunn Index over a finite class of axis-aligned clustering trees with bounded depth, minimum leaf size, and discretized split thresholds. CICT derives prefix-dependent upper bounds by combining an upper bound on achievable inter-cluster separation with lower bounds on the maximum within-cluster diameter that any completion must attain. These diameter bounds use order statistics based on the remaining tree depth and pairs of observations that cannot be separated by any feasible split. The resulting bounds safely prune tree prefixes whose completions cannot improve the incumbent solution. CICT returns an interpretable clustering tree together with a certified residual optimality gap for the prescribed finite tree class. Experiments on benchmark datasets show that CICT achieves competitive regularized Dunn Index values, prunes large portions of the search space, and proves optimality on most tested instances.

1 Introduction

Clustering partitions data into groups that are coherent within groups and separated across groups (Jain, 2010; Ben-David & Ackerman, 2008). In applications where clusters must be inspected or audited, the partition should also admit a simple description. Axis-aligned decision trees provide such descriptions because each observation reaches a leaf through a sequence of feature tests, and each root-to-leaf path defines one cluster (Breiman et al., 1984). Tree depth and number of leaves are common measures of tree complexity and are related to comprehensibility (Piltaver et al., 2016). These properties have motivated optimization methods in which the decision tree directly induces the clustering partition (Bertsimas et al., 2021; Gabidolla & Carreira-Perpiñán, 2022).

We study certified optimization of the regularized Dunn objective over a finite class of clustering trees. Given a fixed dissimilarity matrix, a maximum tree depth D, a minimum leaf size N_C, a regularization parameter τ, and finite threshold sets Θ, the task is to find an axis-aligned decision tree $T \in \mathcal{M}_D(\Theta)$ maximizing $\mathrm{DI}_\tau(T; d)$. The output consists of a feasible tree T_{best}, its objective value, and a residual optimality gap that upper bounds the possible improvement over T_{best} within $\mathcal{M}_D(\Theta)$. A zero gap therefore certifies optimality within this class. The parameters D, N_C, and Θ define the feasible tree class, whereas τ and the dissimilarity d define the objective. The certificate does not extend to continuous thresholds, alternative clustering objectives, perturbation stability, fairness, or downstream performance.

The regularized Dunn objective compares the minimum inter-cluster separation with the maximum within-cluster diameter, using τ as a positive lower bound on the denominator (Dunn, 1974). For a fixed partition, the objective is monotone under cluster-consistent transformations, which contract within-cluster dissimilarities and expand between-cluster dissimilarities (Cohen-Addad et al., 2018). This monotonicity motivates the objective but is distinct from the optimization certificate.

Decision-tree optimization is NP-hard (Bertsimas & Dunn, 2017), and the regularized Dunn objective is non-additive across leaves: splitting a leaf can change the pair that determines the minimum inter-cluster separation or the cluster that determines the maximum within-cluster diameter. This dependence on extreme pairwise dissimilarities makes it difficult to derive bounds that decompose over observations or leaves. Existing optimization-based clustering-tree methods therefore rely primarily on heuristic or local search procedures (Bertsimas et al., 2021; Gabidolla & Carreira-Perpiñán, 2022).

To address this limitation, we introduce **C**ertified **I**nterpretable **C**lustering **T**rees (CICT), a branch-and-bound algorithm for maximizing DI_τ over $\mathcal{M}_D(\Theta)$. The algorithm explores partial trees and computes upper bounds on the best objective value attainable by their feasible completions. Each upper bound combines an upper bound on inter-cluster separation with lower bounds on the maximum within-cluster diameter. The diameter bounds use the remaining tree depth and pairs of observations that cannot be separated by any feasible split.

Contributions. Our main contributions are:

- **Prefix-dependent bounds:** We derive valid upper bounds on the best regularized Dunn value attainable by any feasible completion of a partial tree.
- **Certified search:** We incorporate these bounds into CICT, a branch-and-bound algorithm that returns a feasible clustering tree and a valid residual optimality gap for $\mathcal{M}_D(\Theta)$.
- **Computational evaluation:** We evaluate objective values, runtime, pruning effectiveness, and residual optimality gaps, with comparisons against tree-based and standard clustering baselines.

2 Related Work

Interpretable clustering via trees. Early tree-based clustering methods include decision-tree constructions based on synthetic data (Liu et al., 2000), unsupervised decision trees in the original feature space (Basak & Krishnapuram, 2005), and top-down induction of clustering trees (Blockeel et al., 1998). Clustering Using Unsupervised Binary Trees (CUBT) constructs binary clustering trees through a greedy procedure (Fraiman et al., 2013), with related methods for numerical data (Chen et al., 2006; Gutierrez-Rodríguez et al., 2015) and categorical data (Ghattas et al., 2017).

Post hoc methods first obtain a clustering and then fit a tree to approximate the resulting assignments (Moshkovitz et al., 2020). Intrinsic methods instead optimize the partition induced directly by the tree. ICOT uses mixed-integer formulations for axis-aligned clustering trees and objectives such as the Silhouette and Dunn indices, together with greedy initialization and local improvement (Bertsimas et al., 2021). Clustering trees have also been studied under must-link and cannot-link constraints (Shati et al., 2023), while Tree Alternating Optimization (TAO) has been used to optimize oblique trees (Gabidolla & Carreira-Perpiñán, 2022; Carreira-Perpiñán & Tavallali, 2018).

Other approaches include explainability-first clustering (Hwang & Whang, 2023), soft clustering trees (Cohen, 2023), fuzzy decision-tree clustering (Jiao et al., 2022), decision-tree explanations of kernel clustering (Fleissner et al., 2024), and interpretable neural clustering models (Peng et al., 2022). These methods provide interpretable clustering models or explanations, whereas the present work studies certified optimization of a Dunn-type objective over a finite class of axis-aligned clustering trees.

Optimal supervised decision trees. Mixed-integer programming has been used to optimize supervised decision trees (Bertsimas & Dunn, 2017), and optimization-based tree learning is reviewed by Bertsimas & Dunn (2019). Generalized and Scalable Optimal Sparse Decision Trees (GOSDT) uses branch-and-bound for sparse classification trees (Lin et al., 2020). MurTree combines dynamic programming with search (Demirović et al., 2022), and SParse Lookahead for Interpretable Trees (SPLIT) constructs near-optimal trees with an explicit computational tradeoff (Babbar et al., 2025). Reduced-space branch-and-bound has also been applied to regression trees (Xiong et al., 2025). These methods address supervised objectives that are generally additive over observations or leaves. In contrast, the objective considered here is a ratio determined by extreme pairwise dissimilarities within and across leaves.

Certified optimization and computational bottlenecks. Global optimization methods have been developed for clustering with sum-of-squares objectives (Hua et al., 2021; Chumpitaz-Flores et al., 2025). Their additive structure differs from the minimum-separation to maximum-diameter ratio of the Dunn Index, so their bounds and decomposition methods do not directly apply here. Exact decision-tree methods are commonly evaluated at shallow depths because the number of possible trees grows rapidly (Bertsimas & Dunn, 2017). Similarly, ICOT uses greedy initialization and local improvement rather than exact search over the full tree class (Bertsimas et al., 2021), and optimization-based interpretable clustering methods are generally evaluated on shallow trees or moderate datasets (Gabidolla & Carreira-Perpiñán, 2022; Shati et al., 2023). More broadly, proving optimality can remain difficult on larger instances (Hua et al., 2022), and general-purpose formulations face computational limits when the objective lacks exploitable structure (van der Linden et al., 2023). These limitations motivate objective-specific bounds for the non-additive Dunn objective.

3 Problem Formulation

Let $\mathcal{S} = \{1, \ldots, n\}$ index a dataset with feature vectors $x_i \in [0,1]^P$. Numerical features are rescaled to $[0,1]$. Let $d : \mathcal{S} \times \mathcal{S} \to \mathbb{R}_{\geq 0}$ be a symmetric dissimilarity function satisfying $d(i,i) = 0$. The matrix $\mathrm{dist} \in \mathbb{R}^{n \times n}$ stores the pairwise dissimilarities, with $\mathrm{dist}_{ij} = d(i,j)$.

The task is to partition $\mathcal{S}$ using an axis-aligned binary decision tree. Each internal node applies a split $x_j \leq \theta$, and each leaf defines one cluster. We fix a maximum depth D and a minimum leaf size $N_C \geq 2$. The root has depth zero, and branches may terminate before reaching depth D.

For a tree T, let $\mathcal{L}(T)$ denote its leaves and define

$$\mathcal{S}_t(T) := \{i \in \mathcal{S} : x_i \text{ is routed to leaf } t\}, \qquad t \in \mathcal{L}(T). \tag{1}$$

The partition induced by T is

$$\mathcal{P}_T := \{\mathcal{S}_t(T)\}_{t \in \mathcal{L}(T)}. \tag{2}$$

Every leaf must satisfy the minimum-size constraint

$$|\mathcal{S}_t(T)| \geq N_C, \qquad \forall t \in \mathcal{L}(T). \tag{3}$$

For each feature j, let Θ_j be a finite set of candidate thresholds, and let $\Theta = \{\Theta_j\}_{j=1}^P$. We denote by $\mathcal{M}_D(\Theta)$ the class of axis-aligned trees of depth at most D that satisfy equation 3 and use only thresholds in Θ.

On a finite dataset, thresholds lying between the same pair of consecutive distinct observed values induce the same split. Therefore, one representative threshold between each pair of consecutive distinct values is sufficient to represent every distinct nontrivial axis-aligned split on a feature. Selecting a subset of these representative thresholds reduces the search space and defines a smaller class $\mathcal{M}_D(\Theta)$, as illustrated in Figure 1.

All optimality certificates are relative to $\mathcal{M}_D(\Theta)$. A reference mixed-integer formulation with continuous threshold variables is given in Appendix A. CICT does not solve that formulation; instead it searches over the finite class $\mathcal{M}_D(\Theta)$.

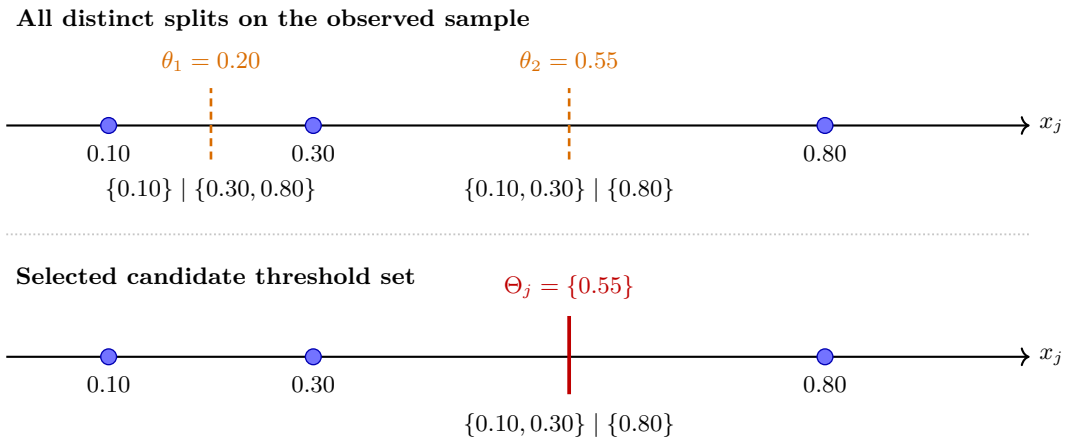
Dissimilarity, diameter, and separation. The Dunn objectives introduced below evaluate the tree-induced partition $\mathcal{P}_T$ through its minimum inter-cluster separation and maximum within-cluster diameter, both defined from the pairwise dissimilarities. For numerical data, Euclidean dissimilarity is computed on the scaled features:

$$d(i,j) = \sqrt{\sum_{k=1}^{P} (x_{ik} - x_{jk})^2}. \tag{4}$$

For any nonempty subset $A \subseteq \mathcal{S}$, define its diameter by

$$\Delta_d(A) := \max_{i,j \in A} d(i,j). \tag{5}$$

Figure 1: Candidate thresholds for a numerical feature with distinct observed values 0.10, 0.30, and 0.80. The upper panel shows one representative threshold for each distinct nontrivial split of the observed sample. The lower panel shows a reduced candidate set containing only the threshold $\theta = 0.55$.



For disjoint nonempty subsets $A, B \subseteq \mathcal{S}$, define their separation by

$$\delta_d(A, B) := \min_{i \in A,\ j \in B} d(i, j). \tag{6}$$

For a partition $\mathcal{P}$ of $\mathcal{S}$, define its maximum within-cluster diameter and, when $|\mathcal{P}| \geq 2$, its minimum inter-cluster separation by

$$\mathrm{diam}_d(\mathcal{P}) := \max_{C \in \mathcal{P}} \Delta_d(C), \tag{7}$$

$$\mathrm{sep}_d(\mathcal{P}) := \min_{\substack{A, B \in \mathcal{P} \\ A \neq B}} \delta_d(A, B). \tag{8}$$

Applied to $\mathcal{P}_T$, these quantities determine the Dunn objectives of the tree T.

3.1 Dunn Index and Regularized Objective

3.1.1 $\mathcal{P}$-consistent transformations

For a fixed reference partition $\mathcal{P}$, a $\mathcal{P}$-consistent transformation contracts within-cluster dissimilarities and expands between-cluster dissimilarities. This property motivates the Dunn objective defined below.

Definition 3.1 ($\mathcal{P}$-consistent transformation)**.** Let $\mathcal{P}$ be a fixed partition of $\mathcal{S}$. A dissimilarity transformation $d \mapsto d'$ is $\mathcal{P}$-consistent if

$$\begin{aligned} d'(i, j) &\leq d(i, j) && \text{if } i \text{ and } j \text{ belong to the same block of } \mathcal{P}, \\ d'(i, j) &\geq d(i, j) && \text{if } i \text{ and } j \text{ belong to different blocks of } \mathcal{P}. \end{aligned}$$

Dunn Index. The Dunn Index is an internal clustering criterion defined as the ratio of the minimum inter-cluster separation to the maximum within-cluster diameter (Dunn, 1974). For a partition $\mathcal{P}$ with at least two blocks and $\mathrm{diam}_d(\mathcal{P}) > 0$, define

$$\mathrm{DI}(\mathcal{P}; d) := \frac{\mathrm{sep}_d(\mathcal{P})}{\mathrm{diam}_d(\mathcal{P})}. \tag{9}$$

If $|\mathcal{P}| = 1$, set $\mathrm{DI}(\mathcal{P}; d) := 0$. The Dunn Index of a tree is the Dunn Index of its induced partition:

$$\mathrm{DI}(T; d) := \mathrm{DI}(\mathcal{P}_T; d). \tag{10}$$

Lemma 3.2 ($\mathcal{P}$-consistent monotonicity of the Dunn Index)**.** *For any fixed partition $\mathcal{P}$ for which the Dunn Index is well defined and any $\mathcal{P}$-consistent transformation d' of d,*

$$\mathrm{DI}(\mathcal{P}; d') \geq \mathrm{DI}(\mathcal{P}; d).$$

The proof is given in Appendix B.1.

3.1.2 Regularized Dunn objective

The Dunn Index can become arbitrarily large when the maximum within-cluster diameter approaches zero. To prevent a near-zero denominator, we optimize the regularized Dunn Index with $\tau > 0$. For a partition $\mathcal{P}$ with at least two blocks, define

$$\mathrm{DI}_\tau(\mathcal{P}; d) := \frac{\mathrm{sep}_d(\mathcal{P})}{\max\{\mathrm{diam}_d(\mathcal{P}), \tau\}}. \tag{11}$$

If $|\mathcal{P}| = 1$, set $\mathrm{DI}_\tau(\mathcal{P}; d) := 0$. For a tree T, define

$$\mathrm{DI}_\tau(T; d) := \mathrm{DI}_\tau(\mathcal{P}_T; d). \tag{12}$$

The parameter τ sets a positive lower bound on the denominator and is fixed throughout the optimization.

Corollary 3.3 ($\mathcal{P}$-consistent monotonicity of the regularized Dunn Index)**.** *For any fixed partition $\mathcal{P}$, any $\tau > 0$, and any $\mathcal{P}$-consistent transformation d' of d,*

$$\mathrm{DI}_\tau(\mathcal{P}; d') \geq \mathrm{DI}_\tau(\mathcal{P}; d).$$

The proof is given in Appendix B.2. This monotonicity motivates the objective for a fixed partition. A separate stability result for the optimizer over the finite class $\mathcal{M}_D(\Theta)$ is given in Appendix B.8.

The certified optimization problem is

$$\max_{T \in \mathcal{M}_D(\Theta)} \mathrm{DI}_\tau(T; d). \tag{13}$$

4 Certified Branch and Bound Algorithm

The objective in equation 11 is non-additive across leaves and depends on worst-case pairs. CICT applies branch and bound to $\mathcal{M}_D(\Theta)$.

Each subproblem is represented by a partial tree π, called a prefix, whose active leaves $\mathcal{L}(\pi)$ form the frontier. A completion of π preserves every leaf at which the STOP action has been selected and may further split only the remaining leaves. A leaf is splittable if its depth is less than D and it admits at least one feasible split.

For a frontier leaf $u \in \mathcal{L}(\pi)$, define

$$\mathcal{S}_u(\pi) := \{i \in \mathcal{S} : x_i \text{ is routed to } u \text{ by } \pi\}.$$

A split (j, θ) is feasible at u if both children satisfy the minimum leaf-size constraint:

$$\left|\{i \in \mathcal{S}_u(\pi) : x_{ij} \leq \theta\}\right| \geq N_C, \qquad \left|\{i \in \mathcal{S}_u(\pi) : x_{ij} > \theta\}\right| \geq N_C. \tag{14}$$

Accordingly, the feasible thresholds for feature j at u are

$$\Theta_j^{\mathrm{feas}}(u; \pi) := \{\theta \in \Theta_j : (j, \theta) \text{ satisfies equation 14 at } u\}. \tag{15}$$

If a threshold is infeasible at u, it remains infeasible at every descendant of u, because every descendant contains a subset of $\mathcal{S}_u(\pi)$. This property is proved in Appendix B.3.

The branching actions at u are

$$\mathcal{A}(\pi, u) := \{\textsc{stop}\} \cup \left\{(j, \theta) : j = 1, \ldots, P,\ \theta \in \Theta_j^{\text{feas}}(u; \pi)\right\}. \tag{16}$$

Let $\mathcal{C}_\Theta(\pi)$ denote the feasible completions of π in $\mathcal{M}_D(\Theta)$. Since D and Θ are finite, $\mathcal{C}_\Theta(\pi)$ is finite whenever it is nonempty, and

$$\mathrm{DI}_\tau^\star(\pi) := \max\{\mathrm{DI}_\tau(T; d) : T \in \mathcal{C}_\Theta(\pi)\}$$

is attained.

The algorithm maintains an incumbent T_{best}, with objective value

$$Q_{\text{best}} := \mathrm{DI}_\tau(T_{\text{best}}; d),$$

and a priority queue $\mathcal{Q}$ of open prefixes ordered by the upper bound $\mathrm{UB}_{\mathrm{DI}_\tau}(\pi)$ defined in Section 5. A prefix is pruned when

$$\mathrm{UB}_{\mathrm{DI}_\tau}(\pi) \le Q_{\text{best}},$$

because its completions cannot improve the incumbent.

Each processed prefix is also completed to obtain a candidate incumbent. Starting from π, the implementation repeatedly applies the feasible split with the largest immediate improvement in DI_τ, and selects STOP when no improving feasible split is available or depth D is reached. This greedy completion affects incumbent quality and runtime, but not the validity of the pruning rule.

Let $U_\mathcal{Q}$ denote the largest upper bound associated with an open prefix, including the incumbent value:

$$U_\mathcal{Q} := \max\left(\{Q_{\text{best}}\} \cup \{\mathrm{UB}_{\mathrm{DI}_\tau}(\pi) : \pi \in \mathcal{Q}\}\right). \tag{17}$$

The residual optimality gap is

$$G := U_\mathcal{Q} - Q_{\text{best}}. \tag{18}$$

If $\mathcal{Q} = \emptyset$, then $U_\mathcal{Q} = Q_{\text{best}}$ and $G = 0$. If the search terminates early, G upper bounds the possible improvement over the incumbent among the remaining open prefixes. The validity of this certificate follows from Proposition 5.4.

The initialization step provides a feasible incumbent whenever $\mathcal{M}_D(\Theta)$ is nonempty. The choice of branching rule affects runtime but not correctness. The method requires the pairwise dissimilarity matrix, which uses $O(n^2)$ memory.

5 Pruning Bound and Certificate

For a tree T, define

$$\mathrm{sep}(T) := \begin{cases} \mathrm{sep}_d(\mathcal{P}_T), & |\mathcal{L}(T)| \ge 2, \\ 0, & |\mathcal{L}(T)| = 1, \end{cases}$$

and

$$\mathrm{diam}(T) := \mathrm{diam}_d(\mathcal{P}_T).$$

Then

$$\mathrm{DI}_\tau(T; d) = \frac{\mathrm{sep}(T)}{\max\{\mathrm{diam}(T), \tau\}}.$$

An upper bound on the objective value of every completion of a prefix is obtained by upper bounding $\mathrm{sep}(T)$ and lower bounding $\mathrm{diam}(T)$.

Algorithm 1 CICT: branch and bound over $\mathcal{M}_D(\Theta)$

Require: distance matrix dist, maximum depth D, minimum leaf size N_C, threshold sets Θ, regularizer τ
Ensure: tree T_{best}, value Q_{best}, residual optimality gap G

```
π_0 ← root prefix
if C_Θ(π_0) = ∅ then
    return infeasible
end if
T_best ← COMPLETE(π_0)
Q_best ← DI_τ(T_best; d)
Q ← {π_0}, keyed by UB_DI_τ
while Q ≠ ∅ and the stopping criterion is not met do
    remove π ∈ Q with largest UB_DI_τ(π)
    if UB_DI_τ(π) ≤ Q_best then
        continue
    end if
    T ← COMPLETE(π)
    if DI_τ(T; d) > Q_best then
        T_best ← T
        Q_best ← DI_τ(T; d)
    end if
    if no splittable leaf exists in π then
        continue
    end if
    choose a splittable leaf u ∈ L(π)
    for each a ∈ A(π, u) do
        π' ← child prefix obtained by applying a at u
        if C_Θ(π') ≠ ∅ and UB_DI_τ(π') > Q_best then
            insert π' into Q
        end if
    end for
end while
compute G from equation 18
return T_best, Q_best, G
```

5.1 Separation Upper Bound

A completion of a prefix can only split its existing leaves; it cannot move observations between distinct prefix leaves. Therefore, any two observations that lie in distinct prefix leaves also lie in distinct leaves of every completion.

For a prefix π with $|\mathcal{L}(\pi)| \geq 2$, define

$$\operatorname{sep}(\pi) := \min_{\substack{u,u' \in \mathcal{L}(\pi) \\ u \neq u'}} \delta_d\big(\mathcal{S}_u(\pi), \mathcal{S}_{u'}(\pi)\big).$$

Proposition 5.1 (Prefix separation upper bound)**.** *Assume $|\mathcal{L}(\pi)| \geq 2$. Then, for every completion $T \in \mathcal{C}_\Theta(\pi)$,*

$$\operatorname{sep}(T) \leq \operatorname{sep}(\pi).$$

The proof is given in Appendix B.4.

5.2 Diameter Lower Bounds

We use two lower bounds on $\mathrm{diam}(T)$. The first depends on the remaining depth below each prefix leaf. The second uses split feasibility to identify pairs of observations that cannot be separated by any feasible completion.

5.2.1 Depth bound via order statistics

For a prefix leaf $u \in \mathcal{L}(\pi)$, let

$$h_u(\pi) := D - \mathrm{depth}(u)$$

denote its remaining depth. Any completion can split the observations reaching u among at most

$$K_u(\pi) := 2^{h_u(\pi)}$$

final leaves. Therefore, at least one final cluster contained in $\mathcal{S}_u(\pi)$ has at least

$$m_u(\pi) := \max\left\{N_C,\ \left\lceil \frac{|\mathcal{S}_u(\pi)|}{K_u(\pi)} \right\rceil\right\}$$

observations.

For each $i \in \mathcal{S}_u(\pi)$, let $\mathrm{NN}_k^{(u,\pi)}(i)$ denote the kth nearest neighbor of i in $\mathcal{S}_u(\pi) \setminus \{i\}$, and define

$$r_i(u;\pi) := d\left(i, \mathrm{NN}_{m_u(\pi)-1}^{(u,\pi)}(i)\right).$$

Let $\mathrm{LB}_{\mathrm{depth}}(u;\pi)$ be the $m_u(\pi)$th order statistic of

$$\{r_i(u;\pi) : i \in \mathcal{S}_u(\pi)\},$$

and define

$$\mathrm{LB}_{\mathrm{depth}}(\pi) := \max_{u \in \mathcal{L}(\pi)} \mathrm{LB}_{\mathrm{depth}}(u;\pi). \tag{19}$$

Proposition 5.2 (Depth-based diameter lower bound)**.** *For every completion $T \in \mathcal{C}_\Theta(\pi)$,*

$$\mathrm{diam}(T) \geq \mathrm{LB}_{\mathrm{depth}}(\pi).$$

The proof is given in Appendix B.5.

5.2.2 Inseparability under feasible splits

The depth-based bound does not use the feasible threshold sets. The second bound uses pairs of observations that cannot be separated by any feasible split below a prefix leaf.

For each $i \in \mathcal{S}_u(\pi)$, define

$$\beta_j^{\mathrm{feas}}(i;u,\pi) := \left|\left\{\theta \in \Theta_j^{\mathrm{feas}}(u;\pi) : x_{ij} > \theta\right\}\right|, \tag{20}$$

$$\beta^{\mathrm{feas}}(i;u,\pi) := \left(\beta_1^{\mathrm{feas}}(i;u,\pi), \ldots, \beta_P^{\mathrm{feas}}(i;u,\pi)\right). \tag{21}$$

If two observations have the same vector $\beta^{\mathrm{feas}}(\cdot;u,\pi)$, then every feasible split at u, or at any descendant of u, sends them to the same child. Consequently, every feasible completion assigns them to the same final cluster.

Define

$$d_{\mathrm{ins}}^{\mathrm{feas}}(u;\pi) := \max\Big\{d(i,i') : i,i' \in \mathcal{S}_u(\pi),\ i \neq i',\ \beta^{\mathrm{feas}}(i;u,\pi) = \beta^{\mathrm{feas}}(i';u,\pi)\Big\}, \tag{22}$$

$$d_{\mathrm{ins}}^{\mathrm{feas}}(\pi) := \max_{u \in \mathcal{L}(\pi)} d_{\mathrm{ins}}^{\mathrm{feas}}(u;\pi), \tag{23}$$

with $d_{\mathrm{ins}}^{\mathrm{feas}}(u;\pi) = 0$ when no distinct pair has the same feasible-threshold vector.

Proposition 5.3 (Diameter lower bound from inseparable pairs)**.** *For every completion $T \in \mathcal{C}_\Theta(\pi)$,*

$$\operatorname{diam}(T) \geq d_{\text{ins}}^{\text{feas}}(\pi).$$

The proof is given in Appendix B.6.

5.3 Optimality Certificate

Combining the two diameter lower bounds gives

$$\text{LB}_{\text{diam}}(\pi) := \max\left\{\text{LB}_{\text{depth}}(\pi), d_{\text{ins}}^{\text{feas}}(\pi)\right\}. \tag{24}$$

For a prefix with a single active leaf u, define

$$d_{\max}(\pi) := \Delta_d\big(\mathcal{S}_u(\pi)\big).$$

The prefix upper bound is

$$\text{UB}_{\text{DI}_\tau}(\pi) := \begin{cases} \dfrac{\operatorname{sep}(\pi)}{\max\{\text{LB}_{\text{diam}}(\pi), \tau\}}, & |\mathcal{L}(\pi)| \geq 2, \\ \dfrac{d_{\max}(\pi)}{\max\{\text{LB}_{\text{diam}}(\pi), \tau\}}, & |\mathcal{L}(\pi)| = 1. \end{cases} \tag{25}$$

Proposition 5.4 (Validity of the prefix upper bound for DI_τ)**.** *For every prefix π with $\mathcal{C}_\Theta(\pi) \neq \emptyset$,*

$$\text{DI}_\tau^\star(\pi) \leq \text{UB}_{\text{DI}_\tau}(\pi).$$

The proof is given in Appendix B.7.

Proposition 5.4 validates the pruning rule in Algorithm 1. Together with equation 17 and equation 18, it also validates the residual optimality gap returned at termination.

Scope of certification. CICT provides an optimality certificate or a residual optimality gap only for the regularized Dunn objective over $\mathcal{M}_D(\Theta)$, with fixed D, N_C, τ, and Θ. Because Θ is part of the prescribed tree class, the certificate does not apply to continuous thresholds or to other clustering objectives.

6 Computational Experiments

The experiments evaluate the certified optimization problem in equation 13. The main outcomes are the regularized Dunn Index DI_τ, the residual optimality gap, runtime, and pruning rate. Silhouette is reported only as a validation metric and is not optimized (Rousseeuw, 1987). The implementation uses `Python 3.10.12`, and all experiments were run on Ubuntu Linux with an Intel® Xeon® Gold 6230R processor, 104 logical cores, 2.10 GHz, and 187 GiB of memory. The datasets are obtained from the UCI Machine Learning Repository (Dua & Graff, 2017), except for Hemicellulose (Wang et al., 2022) and pr2392 (Padberg & Rinaldi, 1991). All features are rescaled to $[0, 1]$, and both DI_τ and silhouette are computed from the same pairwise Euclidean distance matrix.

The comparison methods serve different purposes. Interpretable Clustering via Optimal Trees (ICOT) learns axis-aligned clustering trees and can optimize internal clustering criteria such as the Dunn Index (Bertsimas et al., 2021). Following Gabidolla & Carreira-Perpiñán (2022), Tree Alternating Optimization (TAO) is initialized from a k-means solution and alternates between penalized clustering and oblique-tree updates without using ground-truth labels. Optimal Classification Trees (OCT) is a supervised classification-tree method (Bertsimas & Dunn, 2017); it is trained on class labels for labeled datasets and on k-means pseudo-labels for unlabeled datasets, after which the predicted classes are treated as cluster assignments. The comparison also includes k-means++ (Arthur et al., 2007) and hierarchical clustering. These methods

Table 1: Best DI_τ and silhouette scores across 12 datasets and 6 methods. Bold values indicate the highest score for each dataset and metric. "No solution" indicates that the method did not return a feasible result within the 12-hour time limit. Here, n is the number of observations and P is the number of input features.

DATASET	METHOD	DI_τ	SIL	DATASET	METHOD	DI_τ	SIL	DATASET	METHOD	DI_τ	SIL
IRIS	CICT	**0.3581**	**0.6295**	HCV	CICT	**0.5752**	**0.6267**	HEMI	CICT	**0.3310**	**0.6107**
N = 150	ICOT	**0.3581**	**0.6295**	N = 602	ICOT	0.5145	0.6197	N = 1955	ICOT	**0.3310**	**0.6107**
P = 4	KM++	**0.3581**	**0.6295**	P = 12	KM++	**0.5752**	**0.6267**	P = 7	KM++	0.0296	0.4532
	HCLUST	**0.3581**	**0.6295**		HCLUST	0.4423	0.5082		HCLUST	0.2610	0.5357
	OCT	0.0809	0.4515		OCT	0.0247	-0.1284		OCT	0.0033	0.0401
	TAO	0.2667	0.5802		TAO	0.1011	0.1819		TAO	0.0112	0.2736
SEEDS	CICT	**0.1602**	0.2179	ABS	CICT	**0.4312**	0.2068	PR2392	CICT	**0.3572**	0.2926
N = 210	ICOT	0.1497	0.2574	N = 740	ICOT	**0.4312**	0.2068	N = 2,392	ICOT	**0.3572**	0.2926
P = 7	KM++	0.0966	0.2179	P = 21	KM++	0.4209	**0.2138**	P = 2	KM++	0.0067	0.2982
	HCLUST	0.1228	0.1798		HCLUST	0.3636	0.1983		HCLUST	0.0285	0.3088
	OCT	0.0774	**0.4017**		OCT	0.0123	-0.0494		OCT	0.0076	0.2846
	TAO	0.1188	0.4007		TAO	0.2868	0.2086		TAO	0.0122	**0.3682**
HF_L	CICT	**0.5379**	0.2603	TR	CICT	0.2648	0.2962	WINE	CICT	**0.2612**	0.4706
N = 299	ICOT	**0.5379**	0.2584	N = 980	ICOT	0.1984	**0.3172**	N = 4,898	ICOT	0.2069	0.5024
P = 12	KM++	0.4367	**0.3313**	P = 10	KM++	0.0780	0.1659	P = 7	KM++	0.0146	0.2462
	HCLUST	0.4240	0.1673		HCLUST	**0.2981**	0.2591		HCLUST	0.2288	**0.5117**
	OCT	0.0798	0.0450		OCT	0.0786	0.1443		OCT	NO SOLUTION	
	TAO	0.1782	0.1179		TAO	0.0785	0.1593		TAO	0.0295	0.1465
WHOLE	CICT	**0.6543**	**0.7982**	SGC	CICT	0.3556	0.1923	PAGEBLOCKS	CICT	**0.3110**	0.7635
N = 440	ICOT	**0.6543**	**0.7982**	N = 1,000	ICOT	0.3556	0.1393	N = 5,473	ICOT	0.2291	0.7390
P = 8	KM++	**0.6543**	**0.7982**	P = 21	KM++	0.3244	0.1459	P = 10	KM++	0.0041	0.3625
	HCLUST	**0.6543**	**0.7982**		HCLUST	**0.3957**	**0.2458**		HCLUST	0.1988	0.6764
	OCT	0.0034	-0.2813		OCT	0.1395	0.0498		OCT	NO SOLUTION	
	TAO	0.0688	0.3568		TAO	0.1254	0.0829		TAO	0.1011	**0.8815**

provide reference results, but they do not all optimize the same objective or search over the same tree class. For k-means++ and hierarchical clustering, the number of clusters is set equal to the number of active leaves in the corresponding CICT tree. Their DI_τ and silhouette scores are therefore evaluated using the same number of clusters, although these methods are not constrained to produce tree-induced partitions.

Unless stated otherwise, CICT and ICOT use maximum depth $D = 3$ and minimum leaf size $N_C = 5$. For CICT, the threshold-retention parameter is $\rho = 0.99$, the denominator regularization is $\tau = 0.01 \cdot \mathrm{median}\{d(i,j) : i < j\}$, and the numerical tolerance is $\epsilon_{\mathrm{tol}} = 10^{-4}$. The threshold set Θ is constructed featurewise from sorted unique feature values by retaining thresholds associated with large consecutive gaps according to ρ. Thus, D, N_C, and ρ, through the resulting threshold set Θ, define the finite tree class $\mathcal{M}_D(\Theta)$. The parameter τ defines the regularized objective, while ϵ_{tol} is the numerical tolerance used by the implementation. Because changing ρ changes Θ, it also changes the certified optimization problem. Appendix E.2 reports the sensitivity analysis with respect to ρ.

A method is reported as "no solution" when it does not return a feasible result within the 12-hour time limit. The artifact contains the code for threshold construction, CICT, objective evaluation, and table generation, together with the configurations, preprocessing details, random seeds, stopping criteria, and retained threshold sets.

6.1 Solution Quality and Efficiency

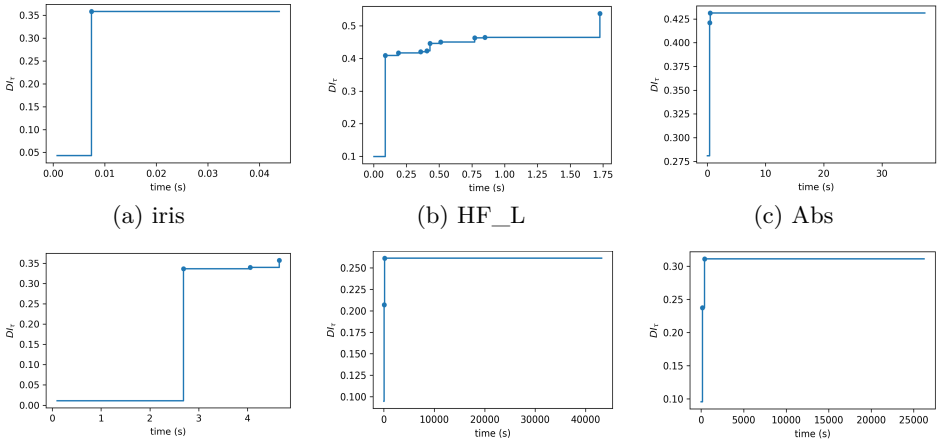
Tables 1 and 2 summarize the solution quality and computational performance of CICT. Within the 12-hour time limit, CICT attains or ties the highest DI_τ on 10 of the 12 datasets while maintaining competitive silhouette scores, including on the larger instances, and certifies a zero residual optimality gap on 11 datasets. Only wine reaches the time limit with a remaining gap of 0.0148. Pruning rates exceed 97% on most datasets, indicating that the proposed upper bounds eliminate a large fraction of the candidate prefixes.

Figures 2 and 3 illustrate the computational behavior and interpretability of CICT, respectively. Figure 2 shows that improved incumbents are often found early, whereas the remaining computation is devoted mainly to pruning prefixes and reducing the residual optimality gap; for wine and pageblocks, the best incumbent

Table 2: Search statistics for CICT. The residual optimality gap G is the upper bound, at termination, on the improvement over the incumbent that remains possible within $\mathcal{M}_D(\Theta)$.

Dataset	Visited	Pruned	Pruning rate (%)	G (↓)	Best DI_τ time (s)	Total time (s)
iris	4	30	88.24	0.0000	0.01	0.03
seeds	129220	4428544	97.16	0.0000	0.07	114.65
HF_L	190	5067	96.39	0.0000	1.69	1.74
whole	26	1903	98.65	0.0000	0.53	2.64
HCV	3	442	99.33	0.0000	1.12	2.45
Abs	2918	126403	97.74	0.0000	0.44	32.89
TR	821	29588	97.30	0.0000	0.95	37.75
SGC	2236	171365	98.71	0.0000	1.62	37.99
hemi	28	3889	99.29	0.0000	4.93	68.39
pr2392	229	9230	97.60	0.0000	2.43	7.63
wine	15977	5073429	99.69	0.0148	50.13	43200
pageblocks	928	595674	99.84	0.0000	398.44	10125.65

Figure 2: Best DI_τ as a function of time. Each curve shows the incumbent trajectory during branch and bound.



(a) iris (b) HF_L (c) Abs

(d) pr2392 (e) wine (f) pageblocks

is obtained before the search terminates. Complementing this search-level view, Figure 3 shows the tree returned for HF_L; leaf labels report cluster sizes, and the splits use binary attributes such as high blood pressure, diabetes, anaemia, and sex.

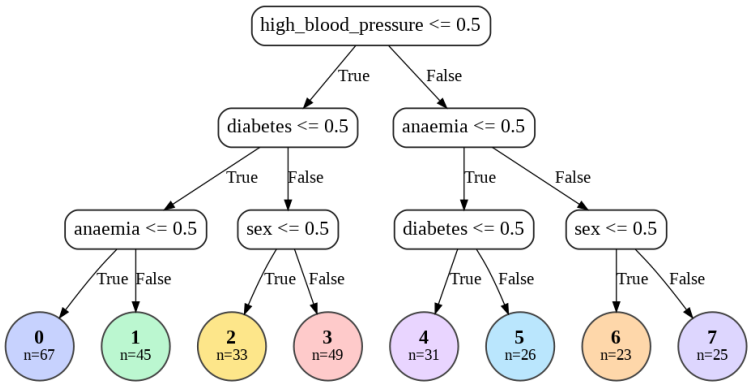
6.2 Perturbation Analysis

The perturbation analysis compares $\mathcal{P}$-consistent perturbations, where $\mathcal{P}$ is the partition returned by CICT for the original distance matrix, with unstructured multiplicative perturbations of the pairwise distances. Throughout these experiments, τ is fixed at the value computed from the original distance matrix. Appendix C reports the complete results. Under deterministic and random $\mathcal{P}$-consistent perturbations, CICT returns partitions with ARI $= 1.0$ on all tested datasets. Under unstructured perturbations, the selected partition can change, particularly for HF_L and Abs. These experiments describe the empirical stability of the returned partitions and are separate from the branch-and-bound optimality certificate over $\mathcal{M}_D(\Theta)$.

7 Discussion

The reported objective values and certificates are conditional on the chosen threshold set and dissimilarity. Because DI_τ depends on the pairwise dissimilarities attaining the minimum inter-cluster separation and the maximum within-cluster diameter, it can be sensitive to individual extreme pairs. The regularization parameter τ prevents a near-zero denominator but does not eliminate this sensitivity. Silhouette provides

Figure 3: Decision tree and induced clusters for HF_L. Each root-to-leaf path defines one cluster.



(a) HF_L

a complementary internal validation criterion (Rousseeuw, 1987), whereas ARI measures agreement with a reference partition when labels are available (Hubert & Arabie, 1985). These criteria assess different properties of a clustering and do not, by themselves, establish performance in downstream tasks (Jain, 2010; Ben-David & Ackerman, 2008).

As is typical in branch-and-bound, a high-quality incumbent may be found well before optimality is certified. For this non-additive objective, loose bounds can make certification substantially more expensive than finding the best incumbent. If the algorithm terminates at a time limit, the residual gap $G > 0$ quantifies the remaining possible improvement within the prescribed tree class. The absolute gap should be reported together with the incumbent value, for example through the relative optimality gap

$$\frac{G}{\max\{\mathrm{DI}_\tau(T_{\mathrm{best}}; d), \varepsilon_{\mathrm{gap}}\}},$$

where $\varepsilon_{\mathrm{gap}} > 0$ avoids division by zero. Scalability is further limited by the $O(n^2)$ storage required for the pairwise dissimilarity matrix and by the combinatorial growth of the search space with tree depth and the number of candidate thresholds. These limitations favor shallow trees and moderate-sized candidate threshold sets.

The returned tree makes cluster assignments inspectable, but it does not provide a causal explanation or establish that the clusters are appropriate for a particular decision. This distinction is especially important when the features contain medical, demographic, financial, or other sensitive information. Sensitive attributes may affect the partition directly, while correlated variables may act as proxies when those attributes are omitted (Barocas et al., 2019; Mehrabi et al., 2021). Robustness analysis, fairness assessment, and domain-specific validation therefore remain necessary before the clusters are used in consequential applications. An optimality certificate establishes solution quality for the stated optimization problem; it does not replace application-specific validation (Rudin, 2019).

8 Conclusion

This work introduced CICT, a branch-and-bound method for maximizing the regularized Dunn objective over a finite class of shallow axis-aligned clustering trees. Its prefix-dependent bounds combine an upper bound on inter-cluster separation with lower bounds on within-cluster diameter derived from order statistics and pairs that cannot be separated by feasible splits, enabling safe pruning and valid residual optimality gaps. Experiments show competitive DI_τ values and zero residual gaps on most tested instances. Thus, CICT returns interpretable tree-induced partitions with certified solution quality over the prescribed finite tree class.

References

David Arthur, Sergei Vassilvitskii, et al. k-means++: The advantages of careful seeding. In *Soda*, volume 7, pp. 1027–1035, 2007.

Varun Babbar, Hayden McTavish, Cynthia Rudin, and Margo Seltzer. Near-optimal decision trees in a SPLIT second. In *Forty-second International Conference on Machine Learning*, 2025. URL `https://openreview.net/forum?id=ACyyBrUioy`.

Solon Barocas, Moritz Hardt, and Arvind Narayanan. *Fairness and Machine Learning: Limitations and Opportunities*. fairmlbook.org, 2019.

Jayanta Basak and Raghu Krishnapuram. Interpretable hierarchical clustering by constructing an unsupervised decision tree. *IEEE Transactions on Knowledge and Data Engineering*, 17(1):121–132, 2005.

Shai Ben-David and Margareta Ackerman. Measures of clustering quality: A working set of axioms for clustering. *Advances in neural information processing systems*, 21, 2008.

Dimitris Bertsimas and Jack Dunn. Optimal classification trees. *Machine Learning*, 106(7):1039–1082, 2017.

Dimitris Bertsimas and Jack Dunn. *Machine learning under a modern optimization lens*. Dynamic Ideas LLC Waltham, 2019.

Dimitris Bertsimas, Agni Orfanoudaki, and Holly Wiberg. Interpretable clustering: an optimization approach. *Machine Learning*, 110(1):89–138, 2021.

Hendrik Blockeel, Luc De Raedt, and Jan Ramon. Top-down induction of clustering trees. In *Proceedings of the Fifteenth International Conference on Machine Learning*, pp. 55–63, 1998.

Leo Breiman, Jerome H. Friedman, Richard A. Olshen, and Charles J. Stone. *Classification and Regression Trees*. Wadsworth, 1984.

Miguel A. Carreira-Perpiñán and Pooya Tavallali. Alternating optimization of decision trees, with application to learning sparse oblique trees. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Yen-Liang Chen, Wei-Han Hsu, and Yu-Hsiang Lee. TASC: Two-attribute-set clustering through decision tree construction. *European Journal of Operational Research*, 174(2):930–944, 2006.

Pedro Chumpitaz-Flores, My Duong, Cristobal Heredia, and Kaixun Hua. A scalable global optimization algorithm for constrained clustering. *arXiv preprint arXiv:2510.22519*, 2025.

Eldan Cohen. Interpretable clustering via soft clustering trees. In *International Conference on Integration of Constraint Programming, Artificial Intelligence, and Operations Research*, pp. 281–298, 2023.

Vincent Cohen-Addad, Varun Kanade, and Frederik Mallmann-Trenn. Clustering redemption–beyond the impossibility of kleinberg's axioms. *Advances in Neural Information Processing Systems*, 31, 2018.

Emir Demirović, Anna Lukina, Emmanuel Hebrard, Jeffrey Chan, James Bailey, Christopher Leckie, Kotagiri Ramamohanarao, and Peter J Stuckey. Murtree: Optimal decision trees via dynamic programming and search. *Journal of Machine Learning Research*, 23(26):1–47, 2022.

Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL `http://archive.ics.uci.edu/ml`.

Joseph C Dunn. Well-separated clusters and optimal fuzzy partitions. *Journal of cybernetics*, 4(1):95–104, 1974.

Maximilian Fleissner, Leena Chennuru Vankadara, and Debarghya Ghoshdastidar. Explaining kernel clustering via decision trees. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=FAGtjl7HOw`.

Ricardo Fraiman, Badih Ghattas, and Marcela Svarc. Interpretable clustering using unsupervised binary trees. *Advances in Data Analysis and Classification*, 7:125–145, 2013.

Magzhan Gabidolla and Miguel Á Carreira-Perpiñán. Optimal interpretable clustering using oblique decision trees. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 400–410, 2022.

Badih Ghattas, Philippe Michel, and Laurent Boyer. Clustering nominal data using unsupervised binary decision trees: Comparisons with the state of the art methods. *Pattern Recognition*, 67:177–185, 2017.

Abraham E. Gutierrez-Rodríguez, J. Francisco Martinez-Trinidad, Milton García-Borroto, and Jesús Ariel Carrasco-Ochoa. Mining patterns for clustering on numerical datasets using unsupervised decision trees. *Knowledge-Based Systems*, 82:70–79, 2015.

Kaixun Hua, Mingfei Shi, and Yankai Cao. A scalable deterministic global optimization algorithm for clustering problems. In *International Conference on Machine Learning*, pp. 4391–4401. PMLR, 2021.

Kaixun Hua, Jiayang Ren, and Yankai Cao. A scalable deterministic global optimization algorithm for training optimal decision tree. *Advances in Neural Information Processing Systems*, 35:8347–8359, 2022.

Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of Classification*, 2:193–218, 1985.

Hyeon Hwang and Steven Euijong Whang. XClusters: Explainability-first clustering. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence*, 2023.

Anil K Jain. Data clustering: 50 years beyond k-means. *Pattern recognition letters*, 31(8):651–666, 2010.

Licheng Jiao, Hong Yang, Zhi-Gang Liu, and Quan Pan. Interpretable fuzzy clustering using unsupervised fuzzy decision trees. *Information Sciences*, 611:540–563, 2022.

Jimmy Lin, Chudi Zhong, Diane Hu, Cynthia Rudin, and Margo Seltzer. Generalized and scalable optimal sparse decision trees. In *International conference on machine learning*, pp. 6150–6160. PMLR, 2020.

Bing Liu, Yiyuan Xia, and Philip S. Yu. Clustering through decision tree construction. In *Proceedings of the Ninth International Conference on Information and Knowledge Management*, pp. 20–29, 2000.

Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *ACM Computing Surveys*, 54(6):1–35, 2021.

Michal Moshkovitz, Sanjoy Dasgupta, Cyrus Rashtchian, and Nave Frost. Explainable k-means and k-medians clustering. In *International conference on machine learning*, pp. 7055–7065. PMLR, 2020.

Manfred Padberg and Giovanni Rinaldi. A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems. *SIAM Review*, 33(1):60–100, 1991.

Xi Peng, Yunfan Li, Ivor W Tsang, Hongyuan Zhu, Jiancheng Lv, and Joey Tianyi Zhou. Xai beyond classification: Interpretable neural clustering. *Journal of Machine Learning Research*, 23(6):1–28, 2022.

Rok Piltaver, Mitja Luštrek, Matjaž Gams, and Sanda Martinčić-Ipšić. What makes classification trees comprehensible? *Expert Systems with Applications*, 62:333–346, 2016.

Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215, 2019.

Pouya Shati, Eldan Cohen, and Sheila A McIlraith. Optimal decision trees for interpretable clustering with constraints. In *IJCAI*, pp. 2022–2030, 2023.

Jacobus van der Linden, Mathijs de Weerdt, and Emir Demirović. Necessary and sufficient conditions for optimal decision trees using dynamic programming. *Advances in Neural Information Processing Systems*, 36:9173–9212, 2023.

Edward Wang, Riley Ballachay, Genpei Cai, Yankai Cao, and Heather L. Trajano. Predicting xylose yield from prehydrolysis of hardwoods: A machine learning approach. *Frontiers in Chemical Engineering*, 4, 2022.

Xuyuan Xiong, Pedro Chumpitaz-Flores, Kaixun Hua, and Cheng Hua. SPOT: Scalable policy optimization with trees for markov decision processes. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL `https://openreview.net/forum?id=OR5WyyTESh`.

A Reference Mixed-Integer Formulation

This appendix gives a reference mixed-integer formulation for tree routing and feasibility with continuous thresholds. Its purpose is to describe the tree-induced partitions and enforce the minimum leaf-size constraint. CICT does not solve this formulation; instead, it searches over the finite discretized class $\mathcal{M}_D(\Theta)$.

We use breadth-first indexing for the nodes of a complete binary tree of maximum depth D. Let

$$N_{\text{nodes}} = 2^{D+1} - 1$$

and define

$$\mathcal{T} = \{1, \dots, N_{\text{nodes}}\}, \qquad \mathcal{T}_B = \left\{1, \dots, \left\lfloor \frac{N_{\text{nodes}}}{2} \right\rfloor \right\},$$

where $\mathcal{T}$ is the set of node positions and $\mathcal{T}_B$ is the set of candidate internal-node positions. The bottom-level node positions are

$$\mathcal{T}_{\text{bot}} = \left\{ \left\lfloor \frac{N_{\text{nodes}}}{2} \right\rfloor + 1, \dots, N_{\text{nodes}} \right\}.$$

For $t \in \mathcal{T} \setminus \{1\}$, let

$$p(t) = \left\lfloor \frac{t}{2} \right\rfloor$$

denote the parent of node t. For each $t \in \mathcal{T}$, let $\mathcal{A}_L(t)$ and $\mathcal{A}_R(t)$ denote the ancestor internal-node positions at which the path from the root to t takes the left and right branch, respectively.

A.1 Tree activation and early stopping

We model the realized tree topology using binary variables

$$u_t,\ s_t,\ \ell_t \in \{0, 1\},$$

where u_t indicates that node t is active, s_t indicates that node t splits, and ℓ_t indicates that node t is an active leaf. The root is always active. At each candidate internal node, an active node either splits or becomes a leaf. At the bottom level, every active node is a leaf. These conditions are encoded by

$$u_1 = 1, \tag{26}$$

$$\ell_t + s_t = u_t, \qquad \forall t \in \mathcal{T}_B, \tag{27}$$

$$\ell_t = u_t, \qquad \forall t \in \mathcal{T}_{\text{bot}}, \tag{28}$$

$$u_{2t} = s_t, \qquad \forall t \in \mathcal{T}_B, \tag{29}$$

$$u_{2t+1} = s_t, \qquad \forall t \in \mathcal{T}_B. \tag{30}$$

Thus, a splitting node activates both children, whereas a node that does not split has no active children.

A.2 Axis-aligned splits

A split at node t selects one feature and one threshold. Let

$$a_{jt} \in \{0, 1\}$$

indicate whether feature j is selected at node t, and define

$$a_t = (a_{1t}, \dots, a_{Pt})^\top.$$

Let $b_t \in [0, 1]$ denote the threshold. We impose

$$\sum_{j=1}^{P} a_{jt} = s_t, \qquad \forall t \in \mathcal{T}_B, \tag{31}$$

$$0 \le b_t \le s_t, \qquad \forall t \in \mathcal{T}_B. \tag{32}$$

When $s_t = 1$, the vector a_t selects exactly one feature. The corresponding split sends observations satisfying

$$a_t^\top x_i \le b_t$$

to the left child and observations satisfying

$$a_t^\top x_i > b_t$$

to the right child. When $s_t = 0$, node t is a leaf and equation 32 forces $b_t = 0$.

A.3 Leaf assignment and minimum leaf size

Let

$$z_{it} \in \{0, 1\}$$

indicate whether observation i is assigned to leaf t. Every observation is assigned to exactly one active leaf:

$$\sum_{t \in \mathcal{T}} z_{it} = 1, \qquad \forall i \in S, \tag{33}$$

$$z_{it} \le \ell_t, \qquad \forall i \in S,\ \forall t \in \mathcal{T}. \tag{34}$$

For a feasible solution, let T denote the realized tree. Its active leaves and their induced clusters are

$$L(T) := \{t \in \mathcal{T} : \ell_t = 1\}, \qquad S_t(T) := \{i \in S : z_{it} = 1\}, \quad t \in L(T),$$

in agreement with the notation used in the main text. The minimum leaf-size constraint is

$$\sum_{i=1}^{n} z_{it} \ge N_C\, \ell_t, \qquad \forall t \in \mathcal{T}. \tag{35}$$

A.4 Path consistency

Assignments must be consistent with every split along the root-to-leaf path. Constant features are removed from the set of admissible split features. After this preprocessing, each indexed feature $j = 1, \ldots, P$ is nonconstant, and we define

$$\varepsilon_j := \min \{|x_{ij} - x_{i'j}| : i, i' \in S,\ x_{ij} \ne x_{i'j}\}.$$

Let

$$\boldsymbol{\varepsilon} = (\varepsilon_1, \ldots, \varepsilon_P)^\top, \qquad \varepsilon_{\max} = \max_{j=1,\ldots,P} \varepsilon_j.$$

For each observation i, candidate leaf t, and ancestor internal node m, impose

$$a_m^\top x_i \le b_m + (1 - z_{it}), \qquad \forall i \in S,\ \forall t \in \mathcal{T},\ \forall m \in \mathcal{A}_L(t), \tag{36}$$

$$a_m^\top x_i \ge b_m + a_m^\top \boldsymbol{\varepsilon} - (1 + \varepsilon_{\max})(1 - z_{it}), \qquad \forall i \in S,\ \forall t \in \mathcal{T},\ \forall m \in \mathcal{A}_R(t). \tag{37}$$

When $z_{it} = 1$, constraint (36) enforces the non-strict left-branch rule. If feature j is selected at node m, constraint (37) becomes

$$x_{ij} \ge b_m + \varepsilon_j.$$

To see why this represents strict right-branch routing on the finite sample, consider any sample partition induced by a continuous threshold. Because every feasible split has observations on both sides, the threshold can be chosen as the largest observed value routed to the left. The smallest distinct observed value routed to the right is then at least ε_j larger. Hence the corresponding right-branch condition is

$$x_{ij} \ge b_m + \varepsilon_j.$$

Thus, the formulation represents all sample partitions induced by continuous thresholds, although different numerical threshold values that induce the same sample partition are not distinguished.

When $z_{it} = 0$, the big-M terms relax the corresponding path constraints.

Constraints (26–37) provide a reference mixed-integer formulation of tree routing and feasibility. They describe, at the observation level, the same type of tree-induced partitions considered in the main text. The certified branch-and-bound algorithm instead searches over the finite discretized class $\mathcal{M}_D(\Theta)$.

B Proofs

B.1 Proof of Lemma 3.2

Proof. The one-block case follows from the convention

$$\mathrm{DI}(\mathcal{P}; d) = 0.$$

Consider a partition $\mathcal{P}$ with at least two blocks for which the unregularized Dunn Index is well defined.

For any distinct blocks $A, B \in \mathcal{P}$,

$$\delta_d(A, B) = \min_{a \in A,\ b \in B} d(a, b).$$

If d' is $\mathcal{P}$-consistent with respect to d, then every between-cluster dissimilarity weakly increases. Therefore,

$$\delta_{d'}(A, B) \geq \delta_d(A, B)$$

for all distinct $A, B \in \mathcal{P}$, and hence

$$\mathrm{sep}_{d'}(\mathcal{P}) \geq \mathrm{sep}_d(\mathcal{P}).$$

Similarly, for every block $C \in \mathcal{P}$,

$$\Delta_d(C) = \max_{x,y \in C} d(x, y).$$

Every within-cluster dissimilarity weakly decreases under a $\mathcal{P}$-consistent transformation, so

$$\Delta_{d'}(C) \leq \Delta_d(C)$$

for all $C \in \mathcal{P}$. Consequently,

$$\mathrm{diam}_{d'}(\mathcal{P}) \leq \mathrm{diam}_d(\mathcal{P}).$$

The numerator of the Dunn Index cannot decrease, while its denominator cannot increase. Therefore,

$$\mathrm{DI}(\mathcal{P}; d') = \frac{\mathrm{sep}_{d'}(\mathcal{P})}{\mathrm{diam}_{d'}(\mathcal{P})} \geq \frac{\mathrm{sep}_d(\mathcal{P})}{\mathrm{diam}_d(\mathcal{P})} = \mathrm{DI}(\mathcal{P}; d).$$

□

B.2 Proof of Corollary 3.3

Proof. The one-block case follows from the convention

$$\mathrm{DI}_\tau(\mathcal{P}; d) = 0.$$

For a partition with at least two blocks, the proof of Lemma 3.2 gives

$$\mathrm{sep}_{d'}(\mathcal{P}) \geq \mathrm{sep}_d(\mathcal{P})$$

and

$$\mathrm{diam}_{d'}(\mathcal{P}) \leq \mathrm{diam}_d(\mathcal{P}).$$

Because $\tau > 0$ is fixed,

$$\max\{\mathrm{diam}_{d'}(\mathcal{P}), \tau\} \leq \max\{\mathrm{diam}_d(\mathcal{P}), \tau\}.$$

Thus,

$$\mathrm{DI}_\tau(\mathcal{P}; d') = \frac{\mathrm{sep}_{d'}(\mathcal{P})}{\max\{\mathrm{diam}_{d'}(\mathcal{P}), \tau\}} \geq \frac{\mathrm{sep}_d(\mathcal{P})}{\max\{\mathrm{diam}_d(\mathcal{P}), \tau\}} = \mathrm{DI}_\tau(\mathcal{P}; d).$$

□

B.3 Feasibility of descendant splits

Lemma B.1 (Feasibility of descendant splits)**.** *Fix a prefix leaf $u \in L(\pi)$ and a completion $T \in \mathcal{C}_\Theta(\pi)$. Let v be an internal descendant of u in T. If T uses split (j, θ) at v, then*

$$\theta \in \Theta_j^{\mathrm{feas}}(u; \pi).$$

Proof. Because v is a descendant of u,

$$S_v(T) \subseteq S_u(\pi).$$

For any $A \subseteq S$, define the left and right child counts associated with (j, θ) by

$$n_L(A) := |\{i \in A : x_{ij} \leq \theta\}|, \qquad n_R(A) := |\{i \in A : x_{ij} > \theta\}|.$$

Since T is feasible and uses (j, θ) at v,

$$n_L(S_v(T)) \geq N_C, \qquad n_R(S_v(T)) \geq N_C.$$

The inclusion $S_v(T) \subseteq S_u(\pi)$ implies

$$n_L(S_u(\pi)) \geq n_L(S_v(T)) \geq N_C$$

and

$$n_R(S_u(\pi)) \geq n_R(S_v(T)) \geq N_C.$$

Thus, the split is feasible at the prefix leaf u. By equation 15,

$$\theta \in \Theta_j^{\mathrm{feas}}(u; \pi).$$

□

B.4 Proof of Proposition 5.1

Proof. Let $u, v \in \mathcal{L}(\pi)$ attain $\mathrm{sep}(\pi)$, and choose

$$i^\star \in \mathcal{S}_u(\pi), \qquad i'^\star \in \mathcal{S}_v(\pi)$$

such that

$$d(i^\star, i'^\star) = \mathrm{sep}(\pi).$$

A completion can only split existing prefix leaves. Therefore, for every $T \in \mathcal{C}_\Theta(\pi)$, the observations $i^\star$ and $i'^\star$ remain in descendants of u and v, respectively.

Let $C_{i^\star}$ and $C_{i'^\star}$ denote their final clusters. These clusters are distinct because $u \neq v$. Hence,

$$\delta_d(C_{i^\star}, C_{i'^\star}) \leq d(i^\star, i'^\star) = \mathrm{sep}(\pi).$$

Since $\mathrm{sep}(T)$ is the minimum separation over all pairs of distinct final clusters,

$$\mathrm{sep}(T) \leq \mathrm{sep}(\pi).$$

□

B.5 Proof of Proposition 5.2

Proof. Fix a prefix leaf $u \in L(\pi)$. A completion can create at most $K_u(\pi)$ final clusters inside $S_u(\pi)$. Therefore, at least one final cluster $C \subseteq S_u(\pi)$ satisfies

$$|C| \geq m_u(\pi).$$

For any $i \in C$, the set $C \setminus \{i\}$ contains at least $m_u(\pi) - 1$ observations from $S_u(\pi) \setminus \{i\}$. Therefore,

$$\max_{i' \in C \setminus \{i\}} d(i, i') \geq r_i(u; \pi).$$

It follows that

$$\Delta_d(C) \geq \max_{i \in C} r_i(u; \pi).$$

Because C contains at least $m_u(\pi)$ observations, this maximum is at least the $m_u(\pi)$-th order statistic of

$$\{r_i(u; \pi) : i \in S_u(\pi)\}.$$

By definition,

$$\Delta_d(C) \geq \mathrm{LB}_{\mathrm{depth}}(u; \pi).$$

Since $\mathrm{diam}(T)$ is the maximum diameter over all final clusters,

$$\mathrm{diam}(T) \geq \mathrm{LB}_{\mathrm{depth}}(u; \pi).$$

Taking the maximum over all prefix leaves yields

$$\mathrm{diam}(T) \geq \mathrm{LB}_{\mathrm{depth}}(\pi).$$

□

B.6 Proof of Proposition 5.3

Proof. Fix a prefix leaf $u \in L(\pi)$, and let $i, i' \in S_u(\pi)$ satisfy

$$\beta^{\mathrm{feas}}(i; u, \pi) = \beta^{\mathrm{feas}}(i'; u, \pi).$$

Consider any completion

$$T \in \mathcal{C}_\Theta(\pi).$$

By Lemma B.1, every split below u uses a threshold in $\Theta_j^{\mathrm{feas}}(u; \pi)$ for its selected feature j.

Equality of the feasible-threshold vectors implies that i and i' lie on the same side of every threshold that can appear below u. Therefore, they take the same branch at every descendant split and belong to the same final cluster C. Consequently,

$$\Delta_d(C) \geq d(i, i').$$

Maximizing first over all pairs with equal feasible-threshold vectors within a prefix leaf and then over all prefix leaves gives

$$\mathrm{diam}(T) \geq d_{\mathrm{ins}}^{\mathrm{feas}}(\pi).$$

□

B.7 Proof of Proposition 5.4

Proof. Fix a completion

$$T \in \mathcal{C}_\Theta(\pi).$$

If $|L(T)| = 1$, then

$$\mathrm{DI}_\tau(T; d) = 0,$$

so the desired inequality holds. Assume henceforth that

$$|L(T)| \geq 2.$$

Suppose first that

$$|L(\pi)| \geq 2.$$

Proposition 5.1 gives

$$\mathrm{sep}(T) \leq \mathrm{sep}(\pi).$$

Propositions 5.2 and 5.3, together with equation 24, give

$$\mathrm{diam}(T) \geq \mathrm{LB}_{\mathrm{diam}}(\pi).$$

Therefore,

$$\begin{aligned}
\mathrm{DI}_\tau(T; d) &= \frac{\mathrm{sep}(T)}{\max\{\mathrm{diam}(T), \tau\}} \\
&\leq \frac{\mathrm{sep}(\pi)}{\max\{\mathrm{LB}_{\mathrm{diam}}(\pi), \tau\}} \\
&= \mathrm{UB}_{\mathrm{DI}_\tau}(\pi).
\end{aligned}$$

Now suppose that

$$|L(\pi)| = 1,$$

and let u be the unique prefix leaf. Every final cluster is contained in $S_u(\pi)$. Hence every pair of observations belonging to distinct final clusters has dissimilarity at most

$$\Delta_d(S_u(\pi)) = d_{\max}(\pi),$$

and therefore

$$\mathrm{sep}(T) \leq d_{\max}(\pi).$$

Using the same diameter lower bound,

$$\mathrm{diam}(T) \geq \mathrm{LB}_{\mathrm{diam}}(\pi),$$

we obtain

$$\mathrm{DI}_\tau(T; d) \leq \frac{d_{\max}(\pi)}{\max\{\mathrm{LB}_{\mathrm{diam}}(\pi), \tau\}} = \mathrm{UB}_{\mathrm{DI}_\tau}(\pi).$$

Thus,

$$\mathrm{DI}_\tau(T; d) \leq \mathrm{UB}_{\mathrm{DI}_\tau}(\pi)$$

for every $T \in \mathcal{C}_\Theta(\pi)$. Therefore, taking the maximum over the finite, nonempty completion set gives

$$\mathrm{DI}^\star_\tau(\pi) = \max_{T \in \mathcal{C}_\Theta(\pi)} \mathrm{DI}_\tau(T; d) \leq \mathrm{UB}_{\mathrm{DI}_\tau}(\pi),$$

which proves the result. □

B.8 Finite-class stability under uniform objective perturbations

Proposition B.2 (Stability under uniform objective perturbations)**.** *Fix the finite tree class $\mathcal{M}_D(\Theta)$, a regularization parameter $\tau > 0$, and a dissimilarity d. For any dissimilarity $\bar{d}$, define*

$$Q_{\bar{d}}(T) := \mathrm{DI}_\tau(T; \bar{d}).$$

Let

$$Q^\star := \max_{T \in \mathcal{M}_D(\Theta)} Q_d(T)$$

and

$$\mathrm{Opt}(d) := \operatorname*{arg\,max}_{T \in \mathcal{M}_D(\Theta)} Q_d(T).$$

If

$$\mathrm{Opt}(d) \neq \mathcal{M}_D(\Theta),$$

define

$$Q^{(2)} := \max_{T \in \mathcal{M}_D(\Theta) \setminus \mathrm{Opt}(d)} Q_d(T)$$

and

$$\gamma := Q^\star - Q^{(2)}.$$

Then $\gamma > 0$. Moreover, if a perturbed dissimilarity d' satisfies

$$\max_{T \in \mathcal{M}_D(\Theta)} |Q_{d'}(T) - Q_d(T)| < \frac{\gamma}{2},$$

then

$$\operatorname*{arg\,max}_{T \in \mathcal{M}_D(\Theta)} Q_{d'}(T) \subseteq \mathrm{Opt}(d).$$

If every tree is optimal under d, the inclusion holds trivially.

Proof. Assume

$$\mathrm{Opt}(d) \neq \mathcal{M}_D(\Theta)$$

and define

$$\xi := \max_{T \in \mathcal{M}_D(\Theta)} |Q_{d'}(T) - Q_d(T)| < \frac{\gamma}{2}.$$

For any $T^\star \in \mathrm{Opt}(d)$,

$$Q_{d'}(T^\star) \geq Q_d(T^\star) - \xi = Q^\star - \xi.$$

For any

$$U \in \mathcal{M}_D(\Theta) \setminus \mathrm{Opt}(d),$$

we have

$$Q_{d'}(U) \leq Q_d(U) + \xi \leq Q^{(2)} + \xi = Q^\star - \gamma + \xi.$$

Since

$$\xi < \frac{\gamma}{2},$$

we obtain

$$Q^\star - \xi > Q^\star - \gamma + \xi.$$

Thus, every tree in $\mathrm{Opt}(d)$ has strictly larger perturbed objective value than every tree outside $\mathrm{Opt}(d)$. Therefore, no tree outside $\mathrm{Opt}(d)$ can maximize $Q_{d'}$. □

C Additional Perturbation Results

Let T_{ref} be the tree returned by CICT for the original dissimilarity d, and let

$$\mathcal{P}^{\mathrm{ref}} := \mathcal{P}_{T_{\mathrm{ref}}}$$

be its induced partition. Throughout these experiments, τ is fixed at the value computed from the original dissimilarity matrix and is not recomputed after perturbation.

We consider three multiplicative perturbation models with perturbation level η.

For the deterministic $\mathcal{P}^{\mathrm{ref}}$-consistent perturbation, define

$$d_\eta^{\mathrm{det}}(i,j) = \begin{cases} (1-\eta)d(i,j), & i,j \text{ belong to the same block of } \mathcal{P}^{\mathrm{ref}}, \\ (1+\eta)d(i,j), & i,j \text{ belong to different blocks of } \mathcal{P}^{\mathrm{ref}}. \end{cases}$$

For the random $\mathcal{P}^{\mathrm{ref}}$-consistent perturbation, each unordered pair $\{i,j\}$ receives an independent multiplicative factor drawn uniformly from

$$[1-\eta, 1]$$

for within-cluster pairs and from

$$[1, 1+\eta]$$

for between-cluster pairs.

For the unstructured perturbation, each unordered pair receives an independent multiplicative factor drawn uniformly from

$$[1-\eta, 1+\eta],$$

without reference to $\mathcal{P}^{\mathrm{ref}}$. In every case, the perturbed matrix is symmetric and has a zero diagonal.

For the random perturbations, we report the mean and standard deviation over seeds

$$\{123, 124, 125\}.$$

Partition agreement is measured using the adjusted Rand index (ARI), which is invariant to permutations of the cluster labels.

Under deterministic $\mathcal{P}^{\mathrm{ref}}$-consistent perturbations, CICT returns the reference partition for every tested dataset and perturbation level, as indicated by ARI $= 1$. The same behavior is observed under the random $\mathcal{P}^{\mathrm{ref}}$-consistent perturbations. ICOT is less stable on HF_L, where its mean ARI decreases as η increases.

The unstructured perturbations serve as a control because they do not preserve the within-cluster and between-cluster ordering associated with $\mathcal{P}^{\mathrm{ref}}$. As η increases, DI_τ generally decreases. The selected partition also changes on HF_L and Abs, where the mean ARI decreases and its variability increases. On iris, HCV, and pr2392, the partition remains unchanged over the tested perturbation levels even though the objective value decreases. These results distinguish the empirical stability observed under cluster-consistent perturbations from the behavior under unstructured perturbations.

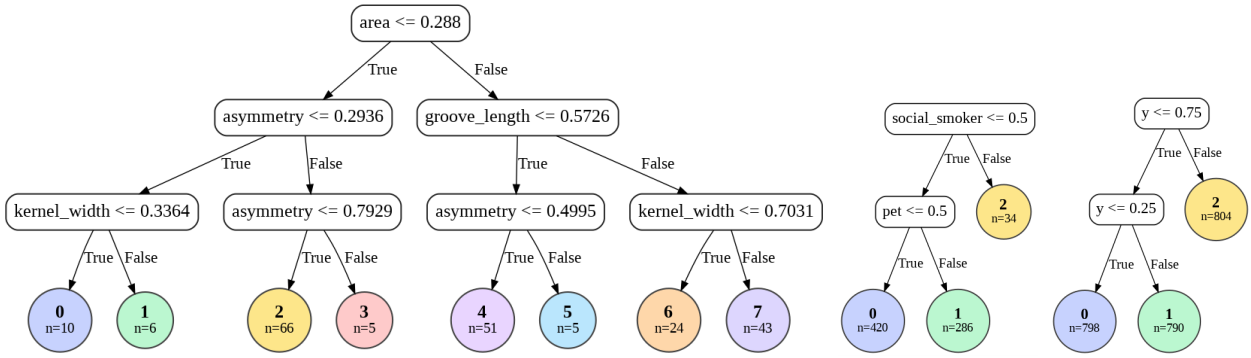
D Additional Tree Visualizations

Figure 4 shows additional trees for the seeds, Abs, and pr2392 datasets. On seeds, the tree uses continuous morphological measurements, including area, asymmetry, kernel-groove length, kernel width, and compactness. On Abs, the tree first splits on social smoker and then on pet. On pr2392, the tree uses coordinate thresholds. These examples complement Tables 1 and 2 by illustrating the root-to-leaf rules that define the returned partitions.

Table 3: Regularized Dunn Index and adjusted Rand index under $\mathcal{P}^{\text{ref}}$-consistent and unstructured dissimilarity perturbations.

Dataset	η	Method	Deterministic $\mathcal{P}^{\text{ref}}$-consistent		Random $\mathcal{P}^{\text{ref}}$-consistent		Unstructured	
			DI_τ	ARI	DI_τ	ARI	DI_τ	ARI
iris $n = 150$ $P = 4$	0.05	CICT	0.3958	1.0000	0.3669 ± 0.0048	1.0000 ± 0.0000	0.3489 ± 0.0123	1.0000 ± 0.0000
		ICOT	0.3958	1.0000	0.3756 ± 0.0035	1.0000 ± 0.0000	0.3481 ± 0.0100	1.0000 ± 0.0000
	0.10	CICT	0.4377	1.0000	0.3760 ± 0.0099	1.0000 ± 0.0000	0.3283 ± 0.0190	1.0000 ± 0.0000
		ICOT	0.4377	1.0000	0.3899 ± 0.0044	1.0000 ± 0.0000	0.3311 ± 0.0126	1.0000 ± 0.0000
	0.20	CICT	0.5372	1.0000	0.3891 ± 0.0151	1.0000 ± 0.0000	0.2840 ± 0.0250	1.0000 ± 0.0000
		ICOT	0.5372	1.0000	0.4158 ± 0.0073	1.0000 ± 0.0000	0.2943 ± 0.0151	1.0000 ± 0.0000
HF_L $n = 299$ $P = 12$	0.05	CICT	0.5945	1.0000	0.5500 ± 0.0010	1.0000 ± 0.0000	0.5117 ± 0.0085	0.9020 ± 0.1697
		ICOT	0.5945	1.0000	0.5518 ± 0.0035	0.9937 ± 0.0109	0.5043 ± 0.0065	0.6877 ± 0.2325
	0.10	CICT	0.6574	1.0000	0.5595 ± 0.0023	1.0000 ± 0.0000	0.4658 ± 0.0080	0.7249 ± 0.1920
		ICOT	0.6574	1.0000	0.5302 ± 0.0550	0.9193 ± 0.1397	0.4692 ± 0.0075	0.5297 ± 0.0923
	0.20	CICT	0.8068	1.0000	0.5674 ± 0.0068	1.0000 ± 0.0000	0.3383 ± 0.0046	0.6619 ± 0.2928
		ICOT	0.8068	1.0000	0.5130 ± 0.1139	0.7542 ± 0.4257	0.4043 ± 0.0071	0.5000 ± 0.1181
HCV $n = 602$ $P = 12$	0.05	CICT	0.6358	1.0000	0.5945 ± 0.0094	1.0000 ± 0.0000	0.5380 ± 0.0030	1.0000 ± 0.0000
		ICOT	0.6358	1.0000	0.5973 ± 0.0089	1.0000 ± 0.0000	0.5481 ± 0.0135	1.0000 ± 0.0000
	0.10	CICT	0.7031	1.0000	0.6069 ± 0.0082	1.0000 ± 0.0000	0.4948 ± 0.0005	1.0000 ± 0.0000
		ICOT	0.7031	1.0000	0.6154 ± 0.0126	1.0000 ± 0.0000	0.5111 ± 0.0186	1.0000 ± 0.0000
	0.20	CICT	0.8629	1.0000	0.6244 ± 0.0030	1.0000 ± 0.0000	0.4122 ± 0.0023	1.0000 ± 0.0000
		ICOT	0.8629	1.0000	0.6460 ± 0.0142	1.0000 ± 0.0000	0.4340 ± 0.0203	1.0000 ± 0.0000
Abs $n = 740$ $P = 21$	0.05	CICT	0.4766	1.0000	0.4416 ± 0.0032	1.0000 ± 0.0000	0.4071 ± 0.0060	0.9423 ± 0.0999
		ICOT	0.4766	1.0000	0.4469 ± 0.0044	1.0000 ± 0.0000	0.4128 ± 0.0031	1.0000 ± 0.0000
	0.10	CICT	0.5271	1.0000	0.4463 ± 0.0017	1.0000 ± 0.0000	0.3781 ± 0.0056	0.8606 ± 0.0592
		ICOT	0.5271	1.0000	0.4590 ± 0.0071	1.0000 ± 0.0000	0.3867 ± 0.0025	0.9548 ± 0.0401
	0.20	CICT	0.6469	1.0000	0.4537 ± 0.0062	1.0000 ± 0.0000	0.3180 ± 0.0096	0.5952 ± 0.3994
		ICOT	0.6469	1.0000	0.4697 ± 0.0064	1.0000 ± 0.0000	0.3332 ± 0.0066	0.9489 ± 0.0302
pr2392 $n = 2{,}392$ $P = 2$	0.05	CICT	0.3948	1.0000	0.3592 ± 0.0010	1.0000 ± 0.0000	0.3271 ± 0.0022	1.0000 ± 0.0000
		ICOT	0.3948	1.0000	0.3616 ± 0.0023	1.0000 ± 0.0000	0.3323 ± 0.0021	1.0000 ± 0.0000
	0.10	CICT	0.4366	1.0000	0.3599 ± 0.0010	1.0000 ± 0.0000	0.2961 ± 0.0021	1.0000 ± 0.0000
		ICOT	0.4366	1.0000	0.3641 ± 0.0033	1.0000 ± 0.0000	0.3063 ± 0.0009	1.0000 ± 0.0000
	0.20	CICT	0.5358	1.0000	0.3611 ± 0.0009	1.0000 ± 0.0000	0.2417 ± 0.0018	1.0000 ± 0.0000
		ICOT	0.5358	1.0000	0.3684 ± 0.0053	1.0000 ± 0.0000	0.2555 ± 0.0017	1.0000 ± 0.0000

Figure 4: Example trees and induced clusters on the seeds, Abs, and pr2392 datasets.



(a) seeds (b) Abs (c) pr2392

E Ablation Studies

E.1 Diameter Lower Bounds

Table 4 compares, with $N_C = 2$ and a 2-hour time limit, three versions of the prefix upper bound: one using only the depth-based order-statistic lower bound $\text{LB}_{\text{depth}}(\pi)$, one using only the inseparable-pair lower bound $d_{\text{ins}}^{\text{feas}}(\pi)$, and one using the combined lower bound

$$\text{LB}_{\text{diam}}(\pi) = \max\left\{\text{LB}_{\text{depth}}(\pi), d_{\text{ins}}^{\text{feas}}(\pi)\right\}.$$

All three variants obtain the same best DI_τ on every dataset, indicating that the bounds primarily affect the search rather than the incumbent objective value.

The inseparable-pair bound alone can lead to substantially more prefix expansions on harder instances such as HF_L and Abs. The combined bound is at least as large as each individual lower bound and generally reduces the number of expanded prefixes. Its wall-clock time need not be the smallest, however, because evaluating both lower bounds introduces additional computation at each prefix.

Table 4: Ablation of the diameter lower bounds under a 2-hour time limit.

Dataset	$\mathrm{LB}_{\mathrm{depth}}(\pi)$				$d_{\mathrm{ins}}^{\mathrm{feas}}(\pi)$				$\mathrm{LB}_{\mathrm{diam}}(\pi)$			
	DI_τ	Visited	Pruned	Time (s)	DI_τ	Visited	Pruned	Time (s)	DI_τ	Visited	Pruned	Time (s)
iris	0.3581	31	219	0.06	0.3581	3	0	0.02	0.3581	3	0	0.03
HCV	0.5752	321	40361	46.31	0.5752	629	18246	43.12	0.5752	123	14436	60.83
HF_L	0.5379	252	9145	1.83	0.5379	214546	929700	160.32	0.5379	252	9145	3.47
Abs	0.4312	5252	239941	38.64	0.4312	3090940	7560384	7212.23	0.4312	5252	239863	52.47
pr2392	0.3572	6	216	5.81	0.3572	6	216	4.53	0.3572	6	216	8.20

E.2 Threshold-Set Construction

Table 5 varies the threshold-retention parameter

$$\rho \in \{0.1, 0.5, 0.9\}.$$

For each value of ρ, the admissible threshold set Θ is constructed before branch-and-bound begins. Thus, CICT certifies the corresponding optimization problem over $\mathcal{M}_D(\Theta)$, and changing ρ changes the finite tree class being searched.

The best DI_τ is unchanged across the tested values of ρ. Larger values of ρ generally reduce the number of visited and pruned prefixes and decrease runtime. On these instances, ρ primarily controls the size of the admissible threshold set and the computational cost of the search.

Table 5: Ablation of the threshold-retention parameter ρ.

Dataset	$\rho = 0.1$				$\rho = 0.5$				$\rho = 0.9$			
	DI_τ	Visited	Pruned	Time (s)	DI_τ	Visited	Pruned	Time (s)	DI_τ	Visited	Pruned	Time (s)
iris	0.3581	73	3393	0.62	0.3581	25	922	0.40	0.3581	4	24	0.02
HCV	0.5752	13	14194	27.23	0.5752	13	9816	19.27	0.5752	3	442	2.70
HF_L	0.5379	192	48738	5.14	0.5379	192	32793	3.36	0.5379	192	5118	0.95
Abs	0.4312	29108	3942688	379.51	0.4312	18526	2067492	219.50	0.4312	2818	122219	26.42
pr2392	0.3572	6	1476	33.42	0.3572	6	916	20.72	0.3572	6	216	5.19

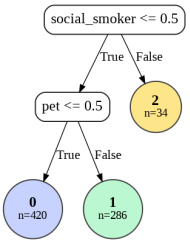
E.3 Maximum Tree Depth

Table 6 reports the effect of the maximum tree depth under a 2-hour time limit. On iris, HCV, and pr2392, the best DI_τ is unchanged as D increases, indicating that additional depth is not needed on these instances. On HF_L, increasing D improves the best objective value. On Abs, the depth-4 class improves over the depth-2 class, whereas the depth-5 search did not complete within the time limit and therefore does not support a direct comparison with the completed depth-4 search.

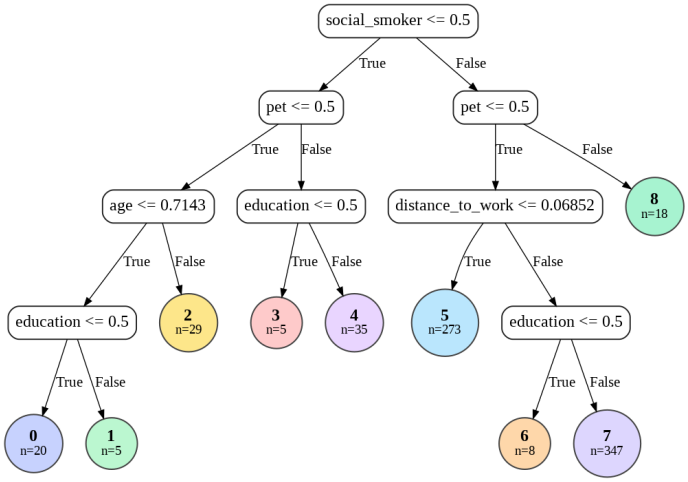
Increasing D enlarges both the admissible tree class and the branch-and-bound search space. Additional depth can improve the objective by permitting further leaf splits, but it can also increase the time required to close the residual optimality gap.

Figure 5 illustrates the effect of the maximum depth on Abs and HF_L. Additional depth permits further splits of existing leaves through additional threshold tests. Such splits can reduce the maximum within-cluster diameter and improve DI_τ. On Abs, the depth-4 tree preserves the principal splits on social smoker and pet, and then further splits the resulting leaves using variables such as age, education, and distance to work. On HF_L, the deeper trees use combinations of binary indicators including high blood pressure, diabetes, anaemia, smoking, and sex.

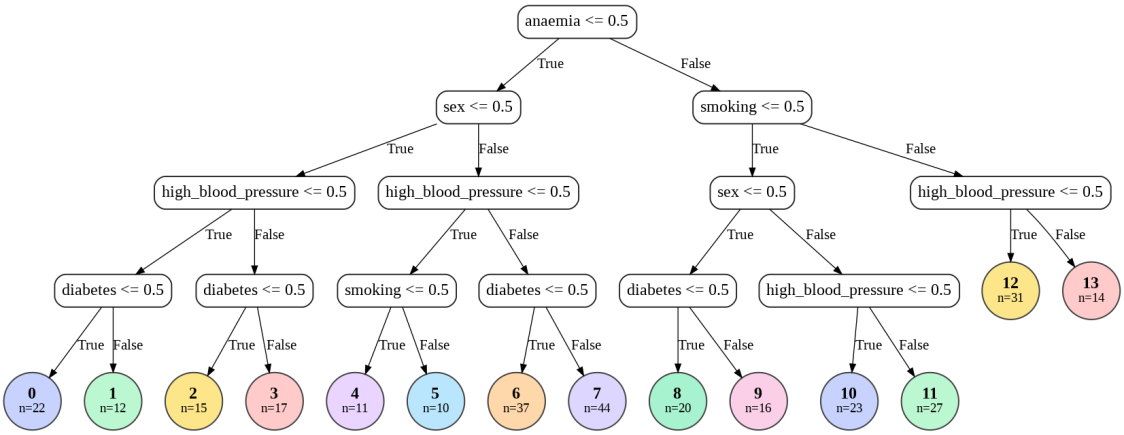
Figure 5: Trees and induced clusters on Abs and HF_L at different maximum depths.



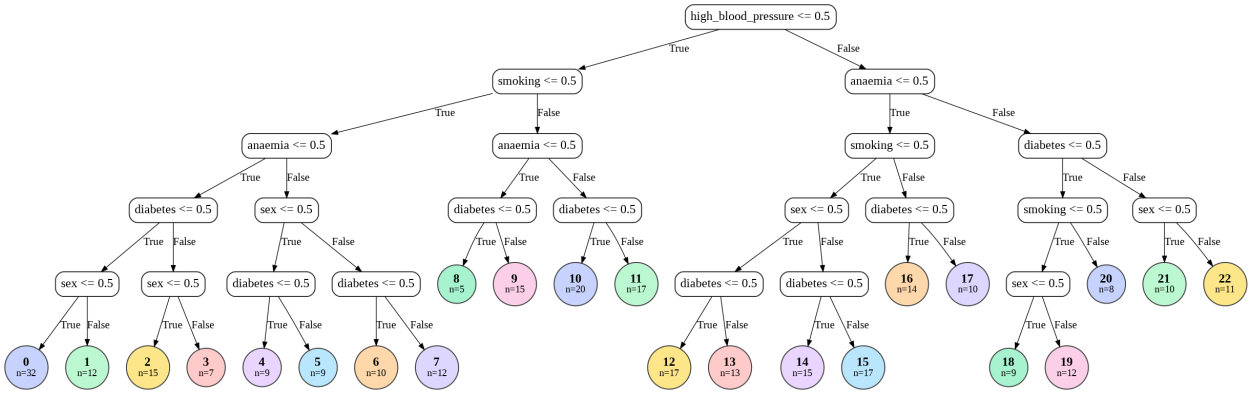
(a) Abs at $D = 2$



(b) Abs at $D = 4$



(c) HF_L at $D = 4$



(d) HF_L at $D = 5$

Table 6: Ablation of the maximum tree depth D.

Dataset	$D = 2$				$D = 4$				$D = 5$			
	DI_τ	Visited	Pruned	Time (s)	DI_τ	Visited	Pruned	Time (s)	DI_τ	Visited	Pruned	Time (s)
iris	0.3581	4	24	0.07	0.3581	4	24	0.07	0.3581	4	24	0.07
HCV	0.5752	3	302	1.43	0.5752	3	442	2.71	0.5752	3	442	2.68
HF_L	0.4647	13	424	0.26	0.6032	13423	214271	15.18	0.7173	38105	399212	47.44
Abs	0.4312	25	1346	3.15	0.4454	1013351	24833717	2764.60	0.4312*	2380609*	40808371*	7220.35*
pr2392	0.3572	6	217	6.47	0.3572	6	216	5.42	0.3572	6	216	5.47

* For Abs at $D = 5$, the search did not complete within the 2-hour time limit. The reported objective is the best incumbent value at termination.

E.4 Minimum Leaf Size

Table 7 reports the effect of varying the minimum leaf size N_C. Increasing N_C removes splits that would produce children with fewer than N_C observations and therefore reduces the admissible search space. This generally decreases the numbers of visited and pruned prefixes and improves runtime. However, a larger minimum leaf size can prevent the tree from isolating small, well-separated groups.

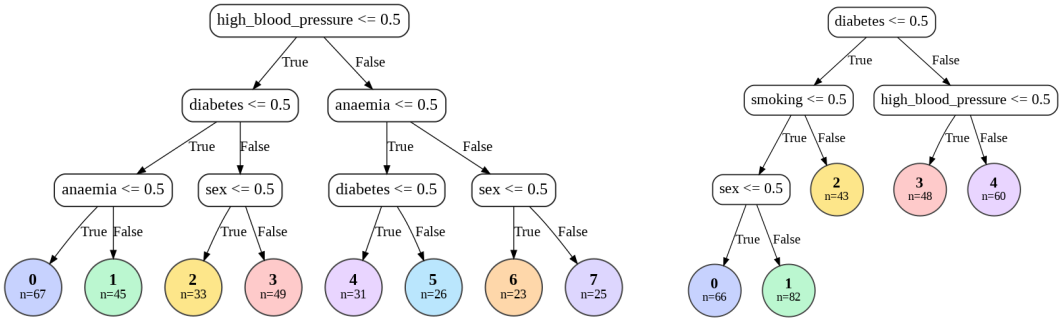
This effect is clearest on HF_L, where the objective is unchanged between $N_C = 2$ and $N_C = 10$, but decreases at $N_C = 40$. HCV and Abs also exhibit smaller objective reductions at $N_C = 40$. On iris and pr2392, the best objective value is unchanged across the tested values, so the stronger constraint primarily reduces the search effort.

Table 7: Ablation of the minimum leaf size N_C.

Dataset	$N_C = 2$				$N_C = 10$				$N_C = 40$			
	DI_τ	Visited	Pruned	Time (s)	DI_τ	Visited	Pruned	Time (s)	DI_τ	Visited	Pruned	Time (s)
iris	0.3581	4	30	0.08	0.3581	4	16	0.07	0.3581	3	0	0.01
HCV	0.5752	123	14818	21.94	0.5752	3	165	1.25	0.5379	3	0	0.19
HF_L	0.5379	252	9145	1.85	0.5379	188	3507	0.70	0.4675	30	113	0.14
Abs	0.4312	5252	239853	36.96	0.4312	1069	38614	13.26	0.4209	35	1064	2.43
pr2392	0.3572	6	216	5.37	0.3572	6	216	5.43	0.3572	6	213	5.56

Figure 6 illustrates the effect of N_C on HF_L. With $N_C = 2$, the tree admits finer splits and can produce several small leaves. With $N_C = 40$, many splits are infeasible because they would create children with fewer than N_C observations. The resulting tree has fewer leaves and coarser clusters. This reduces runtime but can increase the maximum within-cluster diameter and lower DI_τ.

Figure 6: Trees and induced clusters on HF_L at different minimum leaf sizes.



(a) HF_L at $N_C = 2$ (b) HF_L at $N_C = 40$