CAST: LEARNING COMPACT DECISION TREES VIA COMPOUND ADDITIVE SPLITS

Anonymous authors
Paper under double-blind review

ABSTRACT

Decision trees expose their routing rules, but simple splits can require large trees to represent nonlinear boundaries. Nonlinear splits can form richer boundaries, but generic multivariate node functions may obscure how individual features affect routing. We introduce Compound Additive Sparse Trees (CAST), a differentiable single-tree model that forms each split by summing learned univariate feature responses. Node-specific combinations of a shared nonlinear basis and direct linear terms give these responses flexible shapes while preserving an exact feature-wise decomposition of the routing score. Stochastic gates sparsify the feature responses used at each node, and exact folding replaces constant subtrees with leaves. On the complete 20-dataset benchmark, CAST achieves the best mean rank (1.100) in the primary five-method comparison. It ranks first at each matched depth setting from three through six. Plots of selected learned nodes show their nonlinear split boundaries and individual feature contributions in original feature coordinates. Across the primary runs, exact folding removes 30.3% of the starting trees' internal nodes in aggregate while preserving the selected gated trees' predictions.

1 INTRODUCTION

Classical decision trees make predictions along inspectable root-to-leaf paths, with an explicit rule at each internal node. This transparency has made trees a durable model class for tabular data Loh (2011), and tree-based methods remain competitive in modern benchmarks Grinsztajn et al. (2022). Yet complex decision boundaries are often assembled by adding nodes, increasing depth, or combining trees. The resulting structure can make the model harder to inspect as a whole.

A node's splitting rule determines the boundary it can express. An axis-aligned rule thresholds one feature, while an oblique rule thresholds a linear combination of features. Gradient-based methods optimize axis-aligned and oblique rules under a predictive objective Marton et al. (2024); Mao et al. (2026). Their individual split boundaries remain coordinate cuts or hyperplanes. Nonlinear splits can form more flexible boundaries, but when the split uses a generic multivariate function, each feature's effect on routing can be hard to inspect. SGT provides inspectable nonlinear shape-function splits and extends them to two-feature nodes Upadhya & Cohen (2026). Figure 1 contrasts straight cuts with nonlinear neural and additive splits on a two-moons sample.

The additive split in Figure 1(d) sums separate feature responses to route an input. Neural additive models use learned univariate functions to form a prediction Agarwal et al. (2021), and neural basis models share a function basis across features Radenovic et al. (2022). Placing the sum in the routing rule lets several features shape one split while preserving their separate contributions. Composing such splits across nodes allows the tree to represent interactions beyond a single additive score.

We introduce *Compound Additive Sparse Trees* (CAST), a differentiable single-tree model built around this routing rule. CAST forms node-specific nonlinear feature responses from a basis shared across the tree and adds a direct linear term to each response. It learns these responses and the leaf outputs jointly through soft routing. Each feature's contribution to a node's routing score remains separately inspectable, while the complete prediction mixes the leaf outputs. Stochastic gates can remove a feature's complete response from a node, making the active feature set explicit. Exact folding then replaces constant subtrees with leaves while preserving the gated tree's predictions.

In summary, the contributions of this paper are:

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

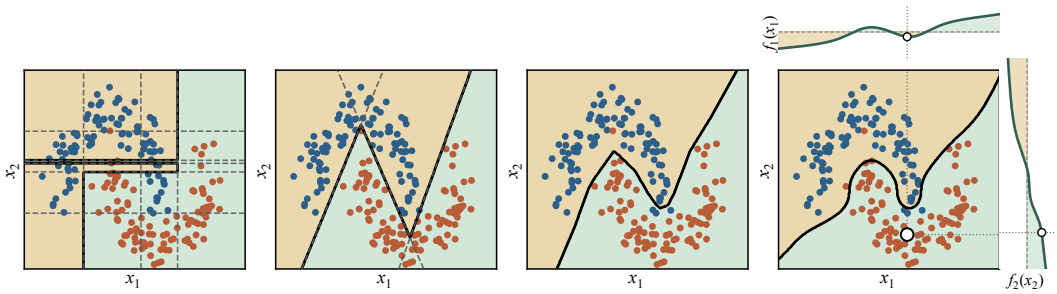


Figure 1: Illustrative split functions on the same standardized two-moons sample. (a) An axis-aligned tree and (b) an oblique tree assemble multiple straight cuts. (c) An MLP forms a curved boundary. (d) A fitted two-feature nonlinear additive split sums separate functions $f_1(x_1)$ and $f_2(x_2)$ with a bias to form its decision score. The upper and right curves show these functions, with outlined points marking their values for the circled sample. Solid black curves mark the resulting decision boundaries. Each gray dashed line indicates one straight split rule in (a) or (b). Yellow and green backgrounds indicate predictions of 0 and 1. Blue and orange points show samples labeled 0 and 1.

1. We develop nonlinear additive splits for a single soft tree. Shared basis functions and node-specific responses let several features shape a boundary while preserving each feature's exact contribution to the routing score.
2. We combine stochastic feature gates with exact subtree folding to simplify the deployed tree. The gates remove complete feature responses from node scores, and folding replaces constant subtrees with leaves to reduce the number of deployed nodes while preserving the gated tree's predictions.
3. We evaluate CAST on the complete 20-dataset benchmark against four other methods. CAST achieves the lowest mean rank in the primary comparison and in four of five matched depth settings. A same-framework linear-node control and deployed-tree analyses examine the nonlinear split's contribution and the resulting structure.

2 Related Work

2.1 Tree Optimization and Differentiable Routing

Classical greedy trees select a feature and threshold at each node using a local criterion (Loh, 2011). Optimal-tree methods search for a whole-tree optimum: GOSDT uses dynamic programming with bounds (Lin et al., 2020), MurTree combines dynamic programming and search (Demirović et al., 2022), and Branches uses AO* search on an AND/OR graph (Chaouki et al., 2025).

Gradient-based tree methods differ in how they handle routing. Soft decision trees replace hard splits with probabilistic routing, allowing a prediction to combine multiple leaves (Irsoy et al., 2012). Deep neural decision forests couple stochastic tree routing with learned neural representations and leaf predictions (Kontschieder et al., 2015). Other methods retain hard routing. GradTree uses a straight-through estimator for axis-aligned trees (Marton et al., 2024). DTSemNet encodes hard oblique trees as neural networks for gradient descent (Panda et al., 2024). RADDT trains hard oblique trees through a ReLU-plus-Argmin reformulation with a softmin approximation for gradient flow (Mao et al., 2026). CAST shares probabilistic routing with deep neural decision forests, but constrains each node score to an additive sum of univariate feature responses.

2.2 Shape-Function Splits

Omnivariate trees select univariate, linear multivariate, or nonlinear multivariate rules at individual nodes (Yildiz & Alpaydin, 2001). BiCART and BiTAO use hard linear splits involving at most two features (Kairgeldin & Carreira-Perpiñán, 2024), while BINN-DT applies nonlinear bivariate

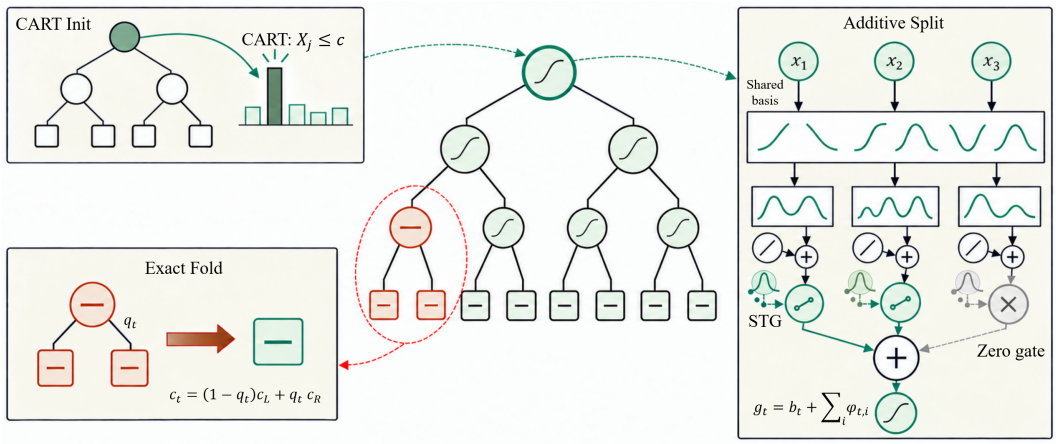


Figure 2: **CAST architecture and training workflow.** A trusted CART root split illustrates the training-only routing initialization (upper left). Each internal node of the soft tree (center) forms a response for each feature from a linear term and a node-specific combination of a shared nonlinear basis (right). Softmax weights and stochastic gates scale these responses before they are summed with the node bias to form the routing score. Gates are perturbed during training and fixed for inference, when a zero gate removes its feature contribution. If every deterministic gate at a node is zero and both child subtrees are constant, exact folding replaces the subtree with one leaf (lower left). Its output is the weighted combination shown, where q_t is the constant right-branch probability. The curves and gate icons illustrate possible node responses and gate states.

splits (Arai et al., 2024). SREDT uses symbolic regression to discover split expressions (Fong & Motani, 2024).

SGT (Upadhya & Cohen, 2026) equips a node with a univariate shape function $f_d(x_d)$, which can assign disjoint intervals of one feature to the same branch. Its ShapeCART procedure constructs a tree greedily using an internal tree to form intervals and a discrete assignment of intervals to branches. The framework also includes TAO refinement and the bivariate S^2GT extension. CAST instead sums one-dimensional feature responses at each node and jointly optimizes these additive scores with soft routing and the leaves.

2.3 Neural Additive Models

Generalized additive models (Hastie, 2017) sum per-feature functions into a prediction. GA^{2}M (Lou et al., 2013) extends additive models with pairwise terms. Neural additive models (Agarwal et al., 2021) learn univariate functions with feature-specific neural networks. Related extensions study distributional regression (Thielmann et al., 2024), prototype-based activation (Xiong et al., 2025), and integrated feature selection (Kishimoto et al., 2024). Neural basis models (NBMs) (Radenovic et al., 2022) share one basis network across features and combine its outputs with feature-specific coefficients, $f_i(x_i) = \mathbf{a}_i^\top \mathbf{h}(x_i)$. Unlike NBMs, which sum feature functions to produce a prediction, CAST sums node-specific feature functions to form the routing score at each tree node.

3 Method

CAST is a soft decision tree with binary branching and compound additive shape splits. A nonlinear basis network is shared across features and nodes, while node-specific parameters form the routing functions. After joint tree training, stochastic-gate continuation can remove complete feature contributions, and exact folding replaces constant subtrees with leaves. Figure 2 summarizes the model architecture and the steps from initialization to exact subtree folding.

3.1 ANCHORED ADDITIVE SPLITS

Let $\mathbf{x} \in \mathbb{R}^p$ denote the input. Let B be the basis dimension. A shared basis $h_\theta : \mathbb{R} \to \mathbb{R}^B$ maps each scalar x_i to a vector of nonlinear basis responses. For node t and feature i, define

$$u_{t,i}(x_i) = \beta_{t,i} x_i + \mathbf{a}_{t,i}^\top h_\theta(x_i), \qquad s_{t,i} = \mathrm{softmax}(\boldsymbol{\pi}_t)_i, \tag{1}$$

where $\beta_{t,i} \in \mathbb{R}$ is a linear anchor, $\mathbf{a}_{t,i} \in \mathbb{R}^B$ contains nonlinear basis coefficients, and $\boldsymbol{\pi}_t$ contains softmax weight logits. With a node–feature gate $m_{t,i} \in [0, 1]$, the node score is

$$g_t(\mathbf{x}; \mathbf{m}) = b_t + \sum_{i=1}^{p} m_{t,i} s_{t,i} u_{t,i}(x_i). \tag{2}$$

The function $\phi_{t,i}(x_i) = m_{t,i} s_{t,i} [\beta_{t,i} x_i + \mathbf{a}_{t,i}^\top h_\theta(x_i)]$ is the exact contribution of feature i to the node score at inference, when $m_{t,i}$ is deterministic. Thus, the score is strictly additive even though each contribution may be nonlinear. The softmax weights remain dense and control the relative scales of the feature responses. A gate fixed at zero for inference removes its feature response from the node score entirely. The shared basis network is learned jointly with the tree.

The linear anchors provide each response with an explicit linear term, while the shared basis adds nonlinear flexibility. The two components are not separately identifiable in general because the nonlinear basis can also represent approximately linear behavior. We therefore interpret and visualize their combined response $\phi_{t,i}$.

The split family includes nonlinear-only additive and oblique linear scores. If $\beta_{t,i} = 0$ for all i, Eq. (2) becomes a nonlinear-only additive split. If $\mathbf{a}_{t,i} = \mathbf{0}$ for all i and the gates are open, it becomes $g_t(\mathbf{x}) = b_t + \sum_i s_{t,i} \beta_{t,i} x_i$. Since softmax gives $s_{t,i} > 0$, any oblique linear score $b_t + \sum_i w_{t,i} x_i$ can be represented by choosing $\beta_{t,i} = w_{t,i} / s_{t,i}$. Proposition 1 in Appendix A.1 shows that a basis network with GELU activations permits a single node to form routing regions that no affine split can form.

3.2 TREE STRUCTURE AND SOFT PREDICTION

The starting depth d specifies a complete tree with $2^d - 1$ internal nodes and 2^d constant leaves before folding. At node t, the probabilities of taking the right and left branches are

$$q_t^{\mathrm{R}}(\mathbf{x}) = \sigma(g_t(\mathbf{x}; \mathbf{m})), \qquad q_t^{\mathrm{L}}(\mathbf{x}) = 1 - q_t^{\mathrm{R}}(\mathbf{x}). \tag{3}$$

For leaf ℓ, let $\mathcal{P}_\ell$ contain the node–edge decisions along its path, and let $\mathcal{L}$ be the current leaf set. Its responsibility and the tree prediction are

$$r_\ell(\mathbf{x}) = \prod_{(t,e) \in \mathcal{P}_\ell} q_t^e(\mathbf{x}), \qquad \hat{y}(\mathbf{x}) = \sum_{\ell \in \mathcal{L}} r_\ell(\mathbf{x}) c_\ell, \tag{4}$$

where c_ℓ is a constant leaf output. Each leaf contains class logits for classification and one scalar for regression. Although each node score is additive, multiplying routing probabilities along a path can introduce interactions between features used at different nodes.

3.3 TRAINING-ONLY TRUSTED ROUTING INITIALIZATION

Before joint optimization, we fit a depth-d CART teacher on part of the training data and check its splits on a disjoint training subset. We transfer a split only when the check supports a positive loss reduction and its ancestors have also been transferred. Each trusted split adjusts the selected feature's softmax logit and linear anchor, together with the node bias, to initialize routing near the teacher threshold. The shared nonlinear basis and node-specific nonlinear coefficients retain their random initialization. CART is discarded after initialization and supplies no prediction at inference. Appendix A.1.2 gives the trust test and parameter updates.

3.4 JOINT TREE TRAINING

During joint tree training, all $m_{t,i} = 1$. The shared basis, linear anchors, nonlinear coefficients, softmax logits, node biases, and leaf outputs are trained together. We minimize cross-entropy for

classification and mean squared error for regression. After routing initialization, leaf outputs are initialized from training examples' soft responsibilities. After joint training, we select the checkpoint with the highest validation accuracy or R^2, computed from the soft leaf mixture, as the starting point for stochastic-gate continuation.

3.5 STOCHASTIC-GATE CONTINUATION

Adapting the stochastic gates of Yamada et al. (2020) to node-feature pairs, we attach a gate parameter $\mu_{t,i}$, initialized to one, to every internal node–model-feature pair of the validation-selected tree. For one Gaussian draw per node–feature pair, shared across examples in a forward pass, with standard deviation σ_{gate}, the training gate is

$$m_{t,i} = \text{clip}(\mu_{t,i} + \epsilon_{t,i}, 0, 1), \quad \epsilon_{t,i} \sim \mathcal{N}(0, \sigma^2_{\text{gate}}). \tag{5}$$

The gate scales the complete linear-plus-shape response after softmax normalization. Closing it does not renormalize the remaining weights or change the input to the shared basis. With T internal nodes, standard-normal CDF Φ, and sparsity weight λ_{gate}, the gate-training objective is

$$\mathcal{L}_{\text{STG}} = \mathbb{E}_{\boldsymbol{\epsilon}}[\mathcal{L}_{\text{pred}}] + \frac{\lambda_{\text{gate}}}{T} \sum_{t=1}^{T} \sum_{i=1}^{p} \Phi(\mu_{t,i}/\sigma_{\text{gate}}). \tag{6}$$

The CDF term is the probability that the noisy gate input is positive, and hence that the clipped gate is nonzero. We continue training the tree and basis parameters together with the gates. The task loss and soft routing remain as in the first stage.

For inference, $m^{\text{det}}_{t,i} = \text{clip}(\mu_{t,i}, 0, 1)$. A gate is closed only when this value is exactly zero. Among checkpoints whose deterministic validation score stays within a task-specific tolerance of the tree-training checkpoint, we select the one with the smallest mean number of exactly nonzero gates per node, breaking ties by validation score and then earlier epoch. Epoch zero restores the complete tree-training checkpoint if no later checkpoint meets this rule. The selected checkpoint is evaluated with deterministic gates and the soft leaf mixture in Eq. (4).

3.6 EXACT CONSTANT-SUBTREE FOLDING

We then inspect the selected tree from the leaves upward. An internal node t is foldable when every deterministic gate at t is exactly zero and both of its child subtrees have already become constant leaves with outputs c_{L} and c_{R}. Its split score is the constant bias b_t. Its right-branch probability is therefore the constant $q_t = \sigma(b_t)$. We replace the entire subtree with the constant output

$$c_t = (1 - q_t)c_{\text{L}} + q_t c_{\text{R}}. \tag{7}$$

For classification, c_{L} and c_{R} are logit vectors mixed before the final class decision. For regression, they are scalar values. Under the local sigmoid routing, constant-leaf, and untruncated soft-mixture configuration, the replacement preserves the subtree's function in exact arithmetic, as formalized in Proposition 2 in Appendix A.1. The resulting artifact is a variable-depth single tree that omits folded nodes and leaves at inference. The folded, executable tree is the CAST predictor used for evaluation. Appendix A.1 gives the full training, checkpoint selection, and exact folding procedure in Algorithm 1.

4 EXPERIMENTS

4.1 DATASETS AND PROTOCOL

We evaluate CAST against differentiable tree methods on our 20-dataset benchmark containing tabular classification and regression tasks. The 11 classification datasets are Adult, Australian, Balance Scale, Bank Marketing, Banknote, Iris, Pendigits, Seismic Bumps, Spambase, Wine, and Wisconsin Breast Cancer. The nine regression datasets are Abalone, Ailerons, California Housing, Concrete, CPU Act, Diabetes, Elevators, Friedman, and Miami Housing 2016. Table 5 in Appendix A.2 reports the task, sample and class counts, feature counts, and split sizes for each dataset. The primary outcomes are classification accuracy and regression R^2, both of which are higher-is-better. We use

seeds 41, 42, and 43, with fixed training, validation, and test partitions for each dataset and seed. All preprocessing is fitted on the training partition. Regression targets are standardized using the training partition during optimization and transformed back to their original scale for evaluation.

4.2 CAST CONFIGURATION

For the primary comparison, CAST starts from one complete tree of depth six with 63 internal nodes and 64 constant leaves. The shared basis network uses GELU activations and has two hidden layers of width 64 with BatchNorm and dropout of 0.1, followed by a 64-dimensional output with dropout of 0.1 during training. The routing temperature is fixed at one, and the linear anchors start at zero before CART initialization. We train the tree for 300 epochs with AdamW (Loshchilov, 2017), learning rate 10^{-3}, weight decay 10^{-4}, a 5% warmup, and cosine decay to 1% of the initial rate. Batch size is 32 for at most 2,000 training examples, 128 for 2,001–10,000 examples, and 256 otherwise. Stochastic-gate continuation runs for 100 epochs without early stopping under a fresh AdamW schedule. Its learning rates are 10^{-4} for the original parameters and 5×10^{-3} for the gates, with weight decay 10^{-4} and zero respectively. The warmup and cosine schedule are the same as in tree training. We use gate-noise standard deviation $\sigma_{\text{gate}} = 0.5$ and sparsity weight $\lambda_{\text{gate}} = 0.01$. We check deterministic validation accuracy or R^2 at epoch zero and every five epochs through epoch 100. A gated checkpoint remains eligible if its score is no more than 0.005 accuracy or 0.01 R^2 below the tree-training checkpoint, according to the task. Appendix A.1.2 gives the CART initialization settings and transfer rule.

4.3 COMPARISON METHODS

We compare CAST with NODE (Popov et al., 2019), RADDT (Mao et al., 2026), GradTree (Marton et al., 2024), and SGT-T (Upadhya & Cohen, 2026). SGT-T is the source paper's designation for a binary univariate SGT built by ShapeCART and refined with ShapeTAO. Its regression evaluation uses the authors' implementation under our protocol. We match methods by their declared depth settings. NODE ensembles oblivious trees, and SGT-T may stop early, so these settings do not fix a common node budget. For each dataset and seed, the methods use the same prepared data and are evaluated on the same test set with the same metric. Appendix A.2.1 summarizes each external method's fitting data, selection rule, and main training or search budget. For aggregate ranks, we first average each method's scores over the three seeds within each dataset. We then rank the five methods on each dataset, assign midranks to ties, and average the 20 dataset ranks. We report dataset-level scores alongside aggregate ranks.

4.4 PRIMARY PREDICTIVE COMPARISON

For each dataset and seed, we apply exact folding to the validation-selected gated tree from Section 3. Folded trees have variable internal-node and leaf counts, while retaining the selected gated tree's soft leaf-mixture predictions up to floating-point error.

Table 1 reports classification accuracy and regression R^2 from the folded CAST trees, evaluated with Eq. (4), and from the four external methods. Entries are three-seed dataset means and sample standard deviations. CAST has the lowest mean rank, 1.100, among the five methods. CAST has a higher three-seed mean than NODE on 19 datasets and ties on one. It has a higher three-seed mean than SGT-T on all 20 datasets.

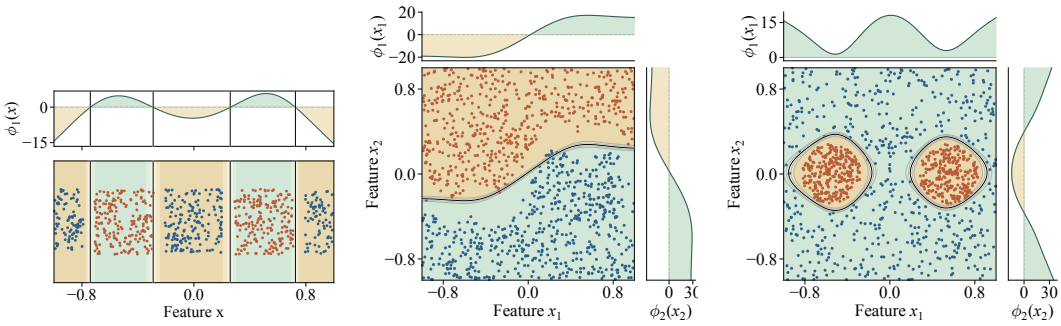
The Friedman test rejects equal ranks among the five methods ($\chi^2(4) = 52.09$, $p = 1.32 \times 10^{-10}$). The subsequent all-pairs Nemenyi test is visualized in Appendix A.2 (Figure 6) (Demšar, 2006). CAST's 1.275-rank gap to NODE does not exceed CD (adjusted $p = 0.080$), whereas its gaps to RADDT, GradTree, and SGT-T do.

4.5 TOPOLOGY AND PREDICTION EQUIVALENCE

For each run in the primary comparison, we pair accuracy or R^2 with the number of internal nodes and leaves in the deployed folded tree. We verify that the original and reloaded folded trees agree on soft predictions within numerical precision and preserve the task metrics. Exact folding removes 1,147 of the 3,780 internal nodes in the 60 starting trees, leaving 2,633 deployed internal nodes, or

Table 1: Depth-six predictive scores on 20 datasets. Entries are three-seed mean ± sample standard deviation of accuracy or R^2, with higher values better. The last row averages ranks of the dataset-level three-seed means, with lower ranks better. Bold and underline mark first and second place by unrounded value. Ties share a rank.

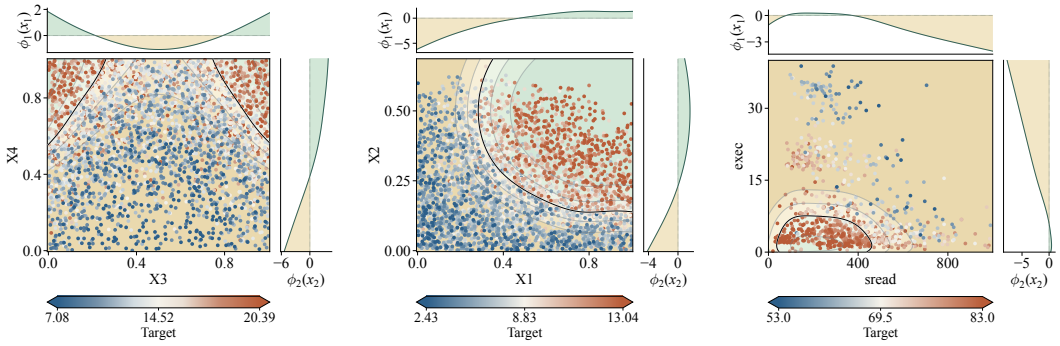
Dataset	CAST	NODE	RADDT	GradTree	SGT-T
Classification: Accuracy					
Adult	**0.8624±0.0018**	0.8607±0.0025	0.8538±0.0027	0.8164±0.0048	0.8601±0.0010
Australian	**0.8599±0.0233**	0.8575±0.0342	0.8068±0.0182	0.8357±0.0221	0.8333±0.0383
Balance Scale	**0.9573±0.0201**	0.9440±0.0349	**0.9573±0.0092**	0.7760±0.0423	0.8107±0.0092
Bank Marketing	**0.7897±0.0095**	0.7861±0.0073	0.7719±0.0049	0.7705±0.0050	0.7799±0.0028
Banknote	0.9976±0.0042	0.9879±0.0021	**1.0000±0.0000**	0.9648±0.0151	0.9648±0.0200
Iris	**0.9556±0.0192**	0.9222±0.0385	0.9333±0.0333	0.9333±0.0577	0.9333±0.0333
Pendigits	**0.9894±0.0013**	0.9886±0.0009	0.9854±0.0024	0.8186±0.0150	0.8992±0.0148
Seismic Bumps	**0.9342±0.0000**	0.9252±0.0081	0.8820±0.0084	0.9329±0.0022	0.9278±0.0062
Spambase	**0.9388±0.0065**	**0.9388±0.0006**	0.9247±0.0140	0.8520±0.0267	0.8907±0.0044
Wine	**0.9907±0.0160**	0.9630±0.0160	0.8981±0.0424	0.9167±0.0278	0.8148±0.0321
Wisconsin Breast Cancer	**0.9643±0.0071**	0.9595±0.0082	0.9310±0.0180	0.9452±0.0180	0.9357±0.0124
Regression: R^2					
Abalone	**0.5660±0.0152**	0.5412±0.0174	0.4544±0.0310	0.4864±0.0125	0.4683±0.0258
Ailerons	**0.8392±0.0055**	0.8233±0.0032	0.7805±0.0072	0.6710±0.0178	0.4654±0.4031
California Housing	**0.7893±0.0109**	0.7736±0.0063	0.7280±0.0110	0.5207±0.0073	0.7046±0.0067
Concrete	**0.8776±0.0234**	0.8694±0.0144	0.8138±0.0170	0.5303±0.0332	0.7234±0.0178
CPU Act	**0.9814±0.0038**	0.9809±0.0026	0.9710±0.0093	0.9439±0.0090	0.9586±0.0047
Diabetes	**0.4923±0.0279**	0.4773±0.0912	0.1756±0.1373	0.3591±0.1085	0.1856±0.1430
Elevators	**0.9099±0.0021**	0.8922±0.0029	0.8958±0.0060	0.4982±0.0102	0.3788±0.3284
Friedman	**0.9580±0.0007**	0.9400±0.0009	0.9126±0.0017	0.6327±0.0036	0.7763±0.0017
Miami Housing 2016	**0.9016±0.0074**	0.8940±0.0039	0.8510±0.0073	0.6550±0.0462	0.7616±0.0172
Mean rank	**1.100**	2.375	3.425	4.125	3.975



(a) Five intervals (b) Wavy boundary (c) Two enclosed regions

Figure 3: Single-node CAST routing on synthetic data. The upper and, for two-feature panels, right curves show the fitted node's exact feature contributions $\phi_i(x_i)$. Black lines mark equal probability of the two child routes, and pale contours show other routing probabilities. Yellow and green backgrounds favor the left and right routes, respectively. Blue and orange points are training samples from classes 0 and 1.

43.88 per tree on average. Exact folding reduces the internal-node count in 54 of the 60 deployed trees. Figure 5 in Appendix A.1 shows the executable topology of one folded tree. Folding preserves the selected gated tree's predictions. Predictive changes arise during training and checkpoint selection. Relative to the validation-selected tree before gate continuation, the final predictor changes mean dataset accuracy by $+0.0071$ and mean dataset R^2 by $+0.0017$ across the 11 classification and nine regression tasks, respectively. A control that continues training from the same checkpoint without stochastic gates has lower selected final mean accuracy by 0.0059 and mean R^2 by 0.0015 (Table 14 in Appendix A.3). This comparison includes checkpoint selection in both pipelines.



(a) Friedman: X_3, X_4 (b) Friedman: X_1, X_2 (c) CPU Act: sread, exec

Figure 4: Feature-wise contributions and conditional routing at three selected CAST nodes. The upper and right curves show each fitted node's two exact contributions $\phi_i(x_i)$. Their sum with the node bias determines the soft route. Black curves mark equal probability of the two child routes, and pale contours show other routing probabilities. Yellow and green backgrounds favor the left and right routes, respectively. Points are training samples colored by target value. The horizontal color bar below each panel gives that panel's target scale. Panels (a) and (b) come from distinct Friedman runs and nodes. Panel (c) uses original CPU activity features and displays $0 \leq \text{sread} \leq 1000$ and $0 \leq \text{exec} \leq 40$.

Table 2: Five-method mean rank by declared depth on 20 datasets. Lower is better. Bold and underline mark first and second place by unrounded rank.

Depth	CAST	NODE	RADDT	GradTree	SGT-T
2	<u>1.875</u>	**1.650**	2.875	4.500	4.100
3	**1.600**	<u>1.850</u>	3.075	4.400	4.075
4	**1.575**	<u>2.175</u>	2.900	4.225	4.125
5	**1.400**	<u>2.225</u>	3.325	4.275	3.775
6	**1.100**	<u>2.375</u>	3.425	4.125	3.975

4.6 Performance across starting depths

Using the same datasets and seeds, we train CAST independently at starting depths two through six and rank all five methods with the procedure above. Table 2 shows that CAST has the lowest mean rank in four of the five matched settings. NODE leads at the smallest depth (1.650 versus CAST's 1.875). Thus, at the same declared depth, CAST achieves stronger aggregate predictive performance in most of the tested settings. Appendix A.2 reports per-dataset scores and CAST's folded node counts, which vary across datasets.

4.7 Contribution of nonlinear node responses

We train a strictly linear-node CAST variant with every nonlinear coefficient fixed at zero, leaving the tree-training protocol and starting topology unchanged. Table 3 compares it with CAST after 300 epochs at each starting depth and at the final prediction endpoint for the primary configuration. The linear variant has lower mean dataset accuracy and R^2 at all five depths, with larger average gaps in shallower trees. This supports nonlinear node responses as a source of predictive capacity under the same starting topology. At the final endpoint, CAST has higher accuracy on ten of eleven classification datasets and higher R^2 on six of nine regression datasets than the linear variant. We also compare complete, nonlinear-only, and linear-only node responses after 300 epochs under a shared initialization without CART transfer (Table 13 in Appendix A.3). Because CART transfer initializes the linear anchor, we disable it in all three arms so that removing the anchor does not also remove a transferred initialization. Relative to the complete response, the nonlinear-only variant changes mean Accuracy by $+0.0019$ and mean R^2 by -0.0060, whereas the linear-only variant changes them by -0.0286 and -0.0164. Under this initialization, the nonlinear-only variant remains close to the complete response in classification, while the linear anchor improves mean regression

Table 3: Strictly linear-node ablation relative to CAST. Entries are mean dataset-level score differences (linear node minus CAST), separately for 11 classification and nine regression datasets. The final row uses the selected predictors after gate continuation and exact folding. Negative values favor CAST.

Evaluation stage	Depth	Δ Accuracy	ΔR^2
After 300 epochs	2	-0.0614	-0.0457
	3	-0.0612	-0.0362
	4	-0.0526	-0.0226
	5	-0.0391	-0.0152
	6	-0.0273	-0.0134
Final predictor	6	-0.0346	-0.0156

R^2. Appendix A.3 also compares CART transfer, basis learning and form, basis width, gate timing, and sparsity weight. At the 300-epoch endpoint, freezing the basis improves mean regression R^2 but lowers classification accuracy (Table 12). Early gate training lowers both final mean scores (Table 14). Recorded training time and sampled memory for these variants appear in Table 15 in the same appendix section.

4.8 Inspecting learned node splits

Figures 3 and 4 show how feature-wise contributions form the routing boundary of an individual CAST node. The synthetic examples isolate the shapes a single node can represent, from repeated intervals in one dimension to curved and enclosed regions in two dimensions. The real-data examples display saved node responses alongside the resulting conditional routing fields. The CPU Act panel uses two original, named input features. Exact feature gates can leave a retained node dependent on only one or two inputs, making its individual response curves and their combined routing boundary directly inspectable. Across the 60 deployed trees in the primary comparison, 477 of 2,633 retained nodes use at most two model-input features (Appendix A.2.2). The figures show the routing geometry of selected nodes.

5 Conclusion

CAST forms nonlinear additive routing rules in a single soft tree using a shared basis and node-specific feature responses. Each feature's contribution to a node score is exact and can be inspected directly. The full prediction remains a soft mixture of leaf outputs.

On the complete 20-dataset benchmark, CAST achieves the lowest mean rank (1.100) among the five compared methods. The same-framework linear-node control supports the predictive value of nonlinear node responses under the same starting topology.

Stochastic gates expose the features used at each node. Exact folding removes 1,147 of 3,780 starting internal nodes across the 60 primary runs, reducing the deployed structure by 30.3%. Together, these results link expressive, inspectable node rules to strong single-tree prediction and a smaller deployed model.

Reproducibility Statement

Section 3 specifies CAST's routing function, training stages, checkpoint selection, and exact folding rule. Section 4 gives the benchmark members, evaluation protocol, and random seeds. Appendix A provides the training algorithm, proofs, dataset characteristics, per-dataset comparisons, and ablation results. Anonymized source code is available at `https://anonymous.4open.science/r/cast_anonymous/`.

AI USE STATEMENT

Generative AI tools were used solely to improve the readability of this manuscript. The authors reviewed all AI-assisted edits and take responsibility for the final content.

REFERENCES

Rishabh Agarwal, Levi Melnick, Nicholas Frosst, Xuezhou Zhang, Ben Lengerich, Rich Caruana, and Geoffrey E Hinton. Neural additive models: Interpretable machine learning with neural nets. *Advances in neural information processing systems*, 34:4699–4711, 2021.

Satoshi Arai, Shinichi Shirakawa, and Tomoharu Nagao. Binn-dt: Towards better interpretability of multidimensional decision rules via bivariate nonlinear node decision trees. In *2024 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 3403–3408. IEEE, 2024.

Ayman Chaouki, Jesse Read, and Albert Bifet. Branches: Efficiently seeking optimal sparse decision trees via ao. 2025.

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.

Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine learning research*, 7(Jan):1–30, 2006.

Kei Sen Fong and Mehul Motani. Symbolic regression enhanced decision trees for classification tasks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 12033–12042, 2024.

Léo Grinsztajn, Edouard Oyallon, and Gaël Varoquaux. Why do tree-based models still outperform deep learning on typical tabular data? *Advances in neural information processing systems*, 35: 507–520, 2022.

Trevor J Hastie. Generalized additive models. In *Statistical models in S*, pp. 249–307. Routledge, 2017.

Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

Ozan Irsoy, Olcay Taner Yıldız, and Ethem Alpaydın. Soft decision trees. In *Proceedings of the 21st international conference on pattern recognition (ICPR2012)*, pp. 1819–1822. IEEE, 2012.

Rasul Kairgeldin and Miguel Á Carreira-Perpiñán. Bivariate decision trees: Smaller, interpretable, more accurate. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 1336–1347, 2024.

Yasutoshi Kishimoto, Kota Yamanishi, Takuya Matsuda, and Shinichi Shirakawa. Neural additive and basis models with feature selection and interactions. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 3–16. Springer, 2024.

Peter Kontschieder, Madalina Fiterau, Antonio Criminisi, and Samuel Rota Bulo. Deep neural decision forests. In *Proceedings of the IEEE international conference on computer vision*, pp. 1467–1475, 2015.

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.

Wei-Yin Loh. Classification and regression trees. *Wiley interdisciplinary reviews: data mining and knowledge discovery*, 1(1):14–23, 2011.

I Loshchilov. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Yin Lou, Rich Caruana, Johannes Gehrke, and Giles Hooker. Accurate intelligible models with pairwise interactions. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 623–631, 2013.

Qiangqiang Mao, Jiayang Ren, Yixiu Wang, Chenxuanyin Zou, Jingjing Zheng, and Yankai Cao. Differentiable decision tree via" relu+ argmin" reformulation. *Advances in Neural Information Processing Systems*, 38:166874–166909, 2026.

Sascha Marton, Stefan Lüdtke, Christian Bartelt, and Heiner Stuckenschmidt. Gradtree: Learning axis-aligned decision trees with gradient descent. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 14323–14331, 2024.

Subrat Prasad Panda, Blaise Genest, Arvind Easwaran, and Ponnuthurai Nagaratnam Suganthan. Vanilla gradient descent for oblique decision trees. In *ECAI 2024: 27th European Conference on Artificial Intelligence, 19–24 October 2024, Santiago de Compostela, Spain–Including 13th Conference on Prestigious Applications of Intelligent Systems (PAIS 2024)*, pp. 1140–1147. SAGE Publications Pvt. Ltd 1 Oliver's Yard, 55 City Road, London, EC1Y 1SP, 2024.

Sergei Popov, Stanislav Morozov, and Artem Babenko. Neural oblivious decision ensembles for deep learning on tabular data. *arXiv preprint arXiv:1909.06312*, 2019.

Filip Radenovic, Abhimanyu Dubey, and Dhruv Mahajan. Neural basis models for interpretability. *Advances in Neural Information Processing Systems*, 35:8414–8426, 2022.

Anton Frederik Thielmann, René-Marcel Kruse, Thomas Kneib, and Benjamin Säfken. Neural additive models for location scale and shape: A framework for interpretable neural regression beyond the mean. In *International conference on artificial intelligence and statistics*, pp. 1783–1791. PMLR, 2024.

Nakul Upadhya and Eldan Cohen. Empowering decision trees via shape function branching. *Advances in Neural Information Processing Systems*, 38:122263–122308, 2026.

Guangzhi Xiong, Sanchit Sinha, and Aidong Zhang. Protonam: Prototypical neural additive models for interpretable deep tabular learning. *ACM Transactions on Knowledge Discovery from Data*, 19(9):1–25, 2025.

Yutaro Yamada, Ofir Lindenbaum, Sahand Negahban, and Yuval Kluger. Feature selection using stochastic gates. In *International conference on machine learning*, pp. 10648–10659. PMLR, 2020.

CT Yildiz and Ethem Alpaydin. Omnivariate decision trees. *IEEE Transactions on Neural Networks*, 12(6):1539–1546, 2001.

A Appendix

A.1 Training Procedure and Properties

A.1.1 Training and inference

Algorithm 1 summarizes the operations that determine the inference predictor. The internal teacher check uses only the training partition, and the teacher is discarded after initialization. The validation-selected tree-training checkpoint is also the epoch-zero fallback for gate selection. The score tolerance for retaining a gated checkpoint is 0.005 accuracy or 0.01 R^2 below that checkpoint, according to the task.

Leaf outputs are initialized from all training examples' soft responsibilities, using responsibility-weighted class frequencies or target means with smoothing toward the training-set prior and small random jitter.

Algorithm 1 Training and exact folding of CAST

Require: Training data D_{tr}, validation data D_{val}, starting depth d
Ensure: Folded soft tree T_{fold}

Initialize the complete depth-d tree and shared basis with all feature gates open.
Fit a depth-d CART teacher on training subset A and check its splits on the disjoint training subset B.
Transfer the trusted, prefix-closed teacher splits and initialize leaf outputs from soft responsibilities on D_{tr}.
Jointly train the tree for 300 epochs and select the best validation soft-prediction checkpoint T_0.
Attach stochastic gates to T_0 and continue training all model parameters for 100 epochs.
Select T_* among the scheduled validation checks and the epoch-zero fallback: retain checkpoints within the task-specific score tolerance of T_0, then minimize the mean number of exactly nonzero gates per node, breaking ties by score and epoch.
Set the selected checkpoint's gates to their deterministic inference values.
for each internal node t of T_* in postorder **do**
    if all deterministic gates at t are exactly zero and both children are constant leaves **then**
        Replace the subtree at t by $[1 - \sigma(b_t)]c_{\mathrm{L}} + \sigma(b_t)c_{\mathrm{R}}$.
    end if
end for
return the executable folded tree T_{fold}.

A.1.2 Trusted routing initialization details

We instantiate the complete tree with a random shared basis and nonlinear coefficients, zero linear anchors and softmax logits, and small random node biases. The teacher uses only the training partition. Within each run, the training examples are split into a teacher-fitting subset A (80%) and an internal check subset B (20%), stratified for classification when feasible.

A depth-d CART teacher is fitted on A with Gini impurity for classification or squared error for regression, a minimum leaf fraction of 0.01, all available features, and the run seed. Parent and child constant predictions are estimated from their respective A regions. For a sample reaching teacher node t, let d_t be the parent prediction loss minus the loss of the child to which that sample routes. These losses are negative log probability for classification and squared error for regression. Class probabilities use unit-count smoothing over all training classes. The predictions used to score B are never refitted on B.

For each teacher node, we bootstrap the reaching B examples 100 times. A split is trusted when the fifth percentile of the resampled means of d_t is positive and both child branches receive at least ten B examples. For a trusted split, its transfer weight is

$$\tau_t = \mathrm{clip}\left(\frac{\mathrm{mean}_{B,t}\, d_t}{\max(\mathrm{mean}_{A,t}\, d_t, 10^{-8})}, 0, 1\right). \tag{8}$$

Otherwise $\tau_t = 0$. Descendants of a zero-weight ancestor retain their original initialization, so transferred splits form a prefix-closed hierarchy.

For a trusted node splitting feature j at threshold c, we update its softmax logit for feature j by setting $\pi_{t,j} \leftarrow (1-\tau_t)\pi_{t,j} + \tau_t \operatorname{logsumexp}(\boldsymbol{\pi}_{t,-j})$ when $p > 1$. The other logits are unchanged. With the resulting softmax weight $s_{t,j}$, define $\kappa_{t,j} = \min\{8, 4/\max(\mathrm{IQR}_A(x_j), 0.5)\}$. We then set $\beta_{t,j} \leftarrow \beta_{t,j} + \tau_t \kappa_{t,j}/s_{t,j}$ and $b_t \leftarrow b_t - \tau_t \kappa_{t,j} c$. For a single feature, $s_{t,j} = 1$. The shared basis and every nonlinear coefficient $\mathbf{a}_{t,i}$ keep their original initialization. Nodes without a trusted split keep all their original node parameters.

A.1.3 A NONLINEAR SPLIT BEYOND AFFINE ROUTING

The linear anchor makes affine scores a subfamily of CAST splits. The learned shared basis gives the split family a strictly larger set of sign patterns, even at a single node.

Proposition 1 (Strict split expressivity)**.** *At deterministic inference, the CAST node-score family contains every affine score on the prepared input. It also contains a one-feature score with two distinct zero crossings. Consequently, its binary routing regions strictly include those obtainable from affine node scores.*

Proof. For an affine score $b + \sum_i w_i x_i$, open the gates, set the nonlinear coefficients to zero, and choose $\beta_i = w_i/s_i$, using $s_i > 0$.

Let $G(z) = z\Phi(z)$ be the GELU activation (Hendrycks & Gimpel, 2016), where Φ is the standard normal CDF. It satisfies $G(z) - G(-z) = z$. At inference, dropout is inactive and the BatchNorm transformations can be set to the identity. Two units in the first hidden layer produce $u = G(x)$ and $v = G(-x)$. Four units in the second hidden layer and one linear output coordinate can then produce

$$[G(u) - G(-u)] + [G(v) - G(-v)] = G(x) + G(-x) = x[2\Phi(x) - 1] = H(x). \tag{9}$$

This uses fewer units than the available width 64. Close every other feature gate and choose the remaining node coefficient and bias so that $g(x) = H(x) - c$ for any $c > 0$. The function H is even, $H(0) = 0$, and it increases from zero to infinity for $x > 0$. Hence g crosses zero once on each side of the origin. A nonconstant affine score restricted to one input coordinate has at most one zero crossing, so it cannot produce this sign pattern. □

The construction characterizes the split family. Which response appears in a fitted tree depends on optimization and checkpoint selection.

A.1.4 EXACT CONSTANT-SUBTREE FOLDING

The folding condition concerns the selected soft predictor with deterministic gates and unit routing temperature.

Proposition 2 (Prediction preservation under exact folding)**.** *Suppose a CAST tree uses local sigmoid routing, untruncated soft leaf mixing, and constant leaf outputs. If every deployed gate at an internal node t is exactly zero and its two child subtrees have become constant leaves with outputs c_{L} and c_{R}, replacing the subtree by Eq.* (7) *leaves the complete tree's output unchanged for every input in exact arithmetic. Any sequence of such replacements in postorder has the same property.*

Proof. The zero gates give $g_t(\mathbf{x}) = b_t$, so the right-route probability $q_t = \sigma(b_t)$ is constant. The subtree output is therefore $(1 - q_t)c_{\mathrm{L}} + q_t c_{\mathrm{R}}$ for every input, equal to the replacement leaf output. Every ancestor receives the same child output after replacement and hence produces the same soft mixture. Induction over the postorder replacements proves the complete-tree statement. For classification, the argument applies to the mixed leaf-logit vector before the final class decision. For regression, it applies to the scalar output. □

Figure 5 illustrates the resulting executable topology for one fitted classification tree.

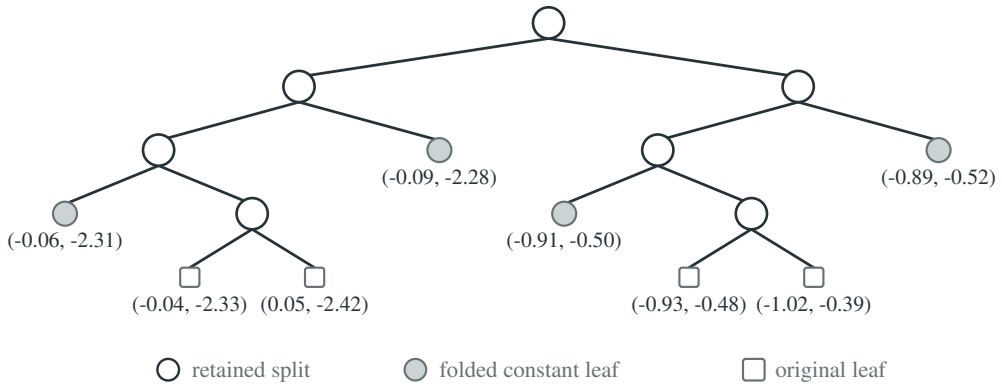


Figure 5: An executable CAST tree after exact folding on the Wisconsin Breast Cancer dataset (seed 43, starting depth four). The selected gated tree has 15 internal nodes and 16 leaves. Its folded form has seven internal nodes and eight leaves. White circles are retained split nodes, gray circles are constant leaves created by folding whole subtrees, and white squares are original leaves. Each pair beneath a leaf gives its stored two-class logit vector, rounded to two decimal places. This tree illustrates how folding preserves the selected soft predictor.

A.2 Dataset Characteristics and Additional Depth Results

Table 5 summarizes the 20 benchmark datasets and their training, validation, and test splits. Tables 6–9 give the five-method comparison at each declared depth. Scores are three-seed mean $\pm$ sample standard deviation of accuracy or R^2. The final row averages ranks of the 20 dataset-level three-seed means. Higher scores and lower ranks are better. Bold and underline mark first and second place by unrounded value, with ties sharing a rank. The final table reports CAST's actual folded topology by starting depth.

A.2.1 External method configurations

Table 4 gives the main fitting, selection, and budget settings used for the external comparisons. Declared depth is matched within each depth table, while each method retains its own training and selection procedure.

Table 4: External comparison configurations per dataset, seed, and declared depth. "Train+val" is the union of the training and validation partitions.

Method	Final fit	Main configuration and budget	Selection
NODE	Train	Classification: two layers of width 1,024. Regression: eight layers of width 128. Up to 20,000 steps, batch 1,024	Validation checkpoint average
RADDT	Train+val	Up to 3,000 epochs, 10 starts and five scales	Best fit objective
GradTree	Train	Up to 1,000 epochs, batch 64, early-stop patience 25	Best validation loss
SGT-T	Train+val	Univariate ShapeCART with ShapeTAO, 50 search attempts	Best completed validation trial

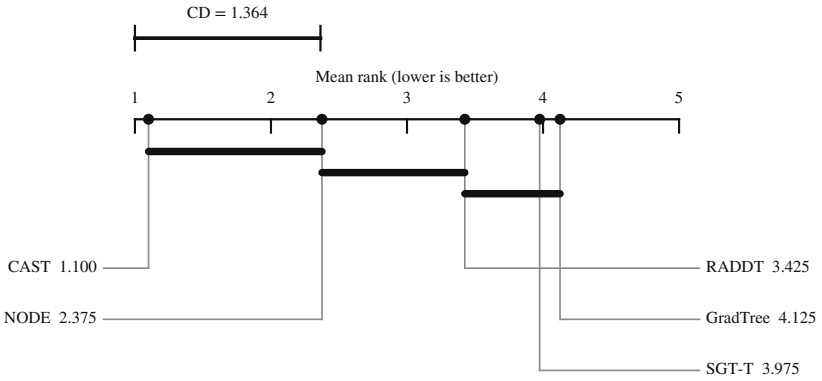


Figure 6: Critical-difference diagram for the depth-six scores in Table 1. The top axis locates each method's mean rank across 20 datasets (lower is better). The upper bracket shows the Nemenyi critical difference (CD) of 1.364 at $\alpha = 0.05$. Thick horizontal bars connect maximal groups whose pairwise rank gaps do not exceed CD. The dataset, rather than the seed, is the repeated-measures unit.

Table 5: Dataset characteristics. Feature counts give raw input columns. Split sizes are train / validation / test.

Dataset	Task	# Samples	# Classes	# Features	Split sizes
Adult	Class.	48,842	2	14	29304 / 9769 / 9769
Australian	Class.	690	2	14	414 / 138 / 138
Balance Scale	Class.	625	3	4	375 / 125 / 125
Bank Marketing	Class.	10,578	2	7	6346 / 2116 / 2116
Banknote	Class.	1,372	2	4	822 / 275 / 275
Iris	Class.	150	3	4	90 / 30 / 30
Pendigits	Class.	10,992	10	16	6594 / 2199 / 2199
Seismic Bumps	Class.	2,584	2	15	1550 / 517 / 517
Spambase	Class.	4,601	2	57	2760 / 920 / 921
Wine	Class.	178	3	13	106 / 36 / 36
Wisconsin Breast Cancer	Class.	699	2	9	419 / 140 / 140
Abalone	Regr.	4,177	–	8	2505 / 836 / 836
Ailerons	Regr.	13,750	–	40	8250 / 2750 / 2750
California Housing	Regr.	20,640	–	8	12384 / 4128 / 4128
Concrete	Regr.	1,030	–	8	618 / 206 / 206
CPU Act	Regr.	8,192	–	21	4914 / 1639 / 1639
Diabetes	Regr.	442	–	10	264 / 89 / 89
Elevators	Regr.	16,599	–	18	9959 / 3320 / 3320
Friedman	Regr.	40,768	–	10	24460 / 8154 / 8154
Miami Housing 2016	Regr.	13,932	–	16	8358 / 2787 / 2787

Table 6: Same-declared-depth predictive scores at depth 2. Bold and underline mark first and second place, respectively.

Dataset	CAST	NODE	RADDT	GradTree	SGT-T
Classification: Accuracy					
Adult	0.8609±0.0023	**0.8623±0.0025**	0.8530±0.0011	0.7781±0.0153	0.8318±0.0005
Australian	0.8551±0.0383	**0.8623±0.0261**	0.8575±0.0425	0.8454±0.0357	0.8454±0.0357
Balance Scale	**0.9787±0.0185**	0.9280±0.0212	0.9173±0.0323	0.6533±0.0568	0.6827±0.0514
Bank Marketing	0.7869±0.0026	**0.7894±0.0081**	0.7544±0.0065	0.7324±0.0116	0.7631±0.0021
Banknote	0.9879±0.0056	0.9879±0.0021	**1.0000±0.0000**	0.9152±0.0056	0.8933±0.0147
Iris	0.8333±0.1453	0.9222±0.0385	**0.9556±0.0192**	0.9000±0.1000	0.9333±0.0333
Pendigits	0.8181±0.0102	**0.9876±0.0017**	0.4067±0.0017	0.3646±0.0277	0.3914±0.0070
Seismic Bumps	**0.9342±0.0000**	0.9310±0.0056	0.9239±0.0107	**0.9342±0.0000**	0.9310±0.0056
Spambase	0.9381±0.0029	**0.9425±0.0047**	0.9182±0.0070	0.7948±0.0210	0.8592±0.0076
Wine	**0.9722±0.0000**	0.9630±0.0160	0.9630±0.0424	0.7870±0.1123	0.8519±0.0699
Wisconsin Breast Cancer	**0.9619±0.0082**	**0.9619±0.0165**	0.9524±0.0109	0.9381±0.0109	0.9238±0.0109
Regression: R^2					
Abalone	**0.5761±0.0189**	0.5333±0.0120	0.5139±0.0318	0.3540±0.0263	0.3516±0.0184
Ailerons	**0.8414±0.0012**	0.8085±0.0038	0.7496±0.0030	0.4142±0.0413	0.1871±0.3242
California Housing	0.7565±0.0115	**0.7733±0.0033**	0.6448±0.0158	0.4208±0.0048	0.4647±0.0120
Concrete	0.8457±0.0208	**0.8636±0.0147**	0.6972±0.0191	0.2774±0.0807	0.4737±0.0498
CPU Act	0.9779±0.0039	**0.9796±0.0026**	0.9405±0.0141	0.8641±0.0386	0.8886±0.0087
Diabetes	0.4276±0.0687	**0.4828±0.0397**	0.3600±0.0494	0.2067±0.1558	0.2247±0.0750
Elevators	**0.9006±0.0021**	0.8869±0.0073	0.8171±0.0044	0.3315±0.0290	0.2502±0.2171
Friedman	**0.9525±0.0028**	0.9434±0.0008	0.6760±0.0064	0.3967±0.0085	0.4159±0.0090
Miami Housing 2016	0.8913±0.0045	**0.8966±0.0038**	0.7040±0.0093	0.4254±0.0367	0.6161±0.0203
Mean rank	1.875	**1.650**	2.875	4.500	4.100

Table 7: Same-declared-depth predictive scores at depth 3. Bold and underline mark first and second place, respectively.

Dataset	CAST	NODE	RADDT	GradTree	SGT-T
Classification: Accuracy					
Adult	0.8580±0.0043	**0.8602±0.0017**	0.8514±0.0016	0.7822±0.0204	0.8528±0.0013
Australian	0.8527±0.0357	**0.8671±0.0293**	0.8213±0.0726	0.8454±0.0357	0.8333±0.0261
Balance Scale	**0.9733±0.0092**	0.9467±0.0333	0.9387±0.0046	0.6880±0.0212	0.7093±0.0395
Bank Marketing	**0.7892±0.0068**	0.7875±0.0090	0.7714±0.0135	0.7391±0.0162	0.7744±0.0039
Banknote	0.9964±0.0036	0.9867±0.0042	**1.0000±0.0000**	0.9661±0.0056	0.9333±0.0172
Iris	0.9222±0.0385	0.9222±0.0385	**0.9556±0.0192**	0.9111±0.0839	0.9444±0.0385
Pendigits	0.9629±0.0086	**0.9886±0.0009**	0.7478±0.0074	0.4823±0.0168	0.6080±0.0043
Seismic Bumps	0.9323±0.0034	0.9336±0.0011	0.9136±0.0087	**0.9342±0.0000**	0.9304±0.0039
Spambase	0.9363±0.0027	**0.9417±0.0051**	0.9146±0.0080	0.8136±0.0101	0.8798±0.0186
Wine	**0.9722±0.0000**	**0.9722±0.0000**	0.9537±0.0321	0.8611±0.1211	0.8611±0.0735
Wisconsin Breast Cancer	**0.9643±0.0071**	0.9571±0.0124	0.9405±0.0149	0.9405±0.0218	0.9262±0.0041
Regression: R^2					
Abalone	**0.5784±0.0215**	0.5416±0.0089	0.5432±0.0281	0.3993±0.0260	0.3970±0.0075
Ailerons	**0.8458±0.0041**	0.8157±0.0055	0.7688±0.0110	0.5153±0.0556	0.6207±0.0110
California Housing	0.7715±0.0111	**0.7727±0.0131**	0.6927±0.0150	0.4343±0.0095	0.5357±0.0014
Concrete	0.8514±0.0273	**0.8710±0.0106**	0.7832±0.0085	0.3969±0.0830	0.5373±0.0063
CPU Act	**0.9802±0.0027**	0.9796±0.0025	0.9625±0.0085	0.8972±0.0301	0.9267±0.0058
Diabetes	**0.4711±0.0353**	0.4679±0.0783	0.3475±0.0521	0.3252±0.1073	0.2184±0.1958
Elevators	**0.9081±0.0023**	0.8911±0.0041	0.8773±0.0035	0.4166±0.0127	0.1450±0.2519
Friedman	**0.9585±0.0011**	0.9417±0.0005	0.7708±0.0022	0.4479±0.0591	0.5869±0.0086
Miami Housing 2016	**0.8973±0.0098**	0.8938±0.0002	0.8027±0.0029	0.5229±0.0327	0.6831±0.0217
Mean rank	**1.600**	1.850	3.075	4.400	4.075

Table 8: Same-declared-depth predictive scores at depth 4. Bold and underline mark first and second place, respectively.

Dataset	CAST	NODE	RADDT	GradTree	SGT-T
Classification: Accuracy					
Adult	0.8607±0.0030	**0.8623±0.0026**	0.8493±0.0024	0.8076±0.0005	0.8595±0.0023
Australian	0.8527±0.0357	0.8406±0.0316	0.8213±0.0293	**0.8575±0.0151**	0.8213±0.0302
Balance Scale	**0.9840±0.0160**	0.9493±0.0323	0.9787±0.0201	0.7760±0.0400	0.7947±0.0122
Bank Marketing	0.7864±0.0038	**0.7883±0.0037**	0.7727±0.0086	0.7647±0.0140	0.7678±0.0224
Banknote	0.9976±0.0042	0.9867±0.0042	**1.0000±0.0000**	0.9600±0.0063	0.9624±0.0076
Iris	0.9333±0.0333	0.9222±0.0385	**0.9556±0.0192**	0.9444±0.0509	0.9333±0.0333
Pendigits	0.9798±0.0021	**0.9882±0.0023**	0.9612±0.0003	0.6382±0.0698	0.7884±0.0094
Seismic Bumps	**0.9342±0.0000**	0.9291±0.0059	0.9052±0.0039	**0.9342±0.0000**	0.9291±0.0073
Spambase	0.9352±0.0027	**0.9414±0.0038**	0.9276±0.0070	0.8266±0.0115	0.8925±0.0179
Wine	**0.9722±0.0000**	**0.9722±0.0000**	0.9167±0.0278	0.9074±0.0160	0.8241±0.0160
Wisconsin Breast Cancer	**0.9619±0.0041**	0.9571±0.0189	0.9571±0.0071	0.9405±0.0041	0.9286±0.0189
Regression: R^2					
Abalone	**0.5735±0.0061**	0.5430±0.0111	0.5527±0.0141	0.4035±0.0377	0.4593±0.0151
Ailerons	**0.8402±0.0092**	0.8199±0.0058	0.7655±0.0135	0.6334±0.0109	0.4482±0.3882
California Housing	**0.7778±0.0059**	0.7694±0.0086	0.7148±0.0115	0.4883±0.0128	0.6314±0.0100
Concrete	0.8586±0.0099	**0.8694±0.0283**	0.7970±0.0248	0.4384±0.0741	0.6097±0.0637
CPU Act	**0.9809±0.0025**	0.9789±0.0037	0.9590±0.0137	0.9314±0.0077	0.9457±0.0062
Diabetes	0.4822±0.0276	**0.4906±0.0720**	0.2735±0.1585	0.3433±0.0696	0.0756±0.0202
Elevators	**0.9101±0.0021**	0.8909±0.0044	0.8933±0.0025	0.4424±0.0269	0.4942±0.0230
Friedman	**0.9592±0.0007**	0.9406±0.0008	0.8352±0.0010	0.5513±0.0362	0.6473±0.0059
Miami Housing 2016	**0.9000±0.0079**	0.8978±0.0027	0.8349±0.0101	0.6291±0.0239	0.7407±0.0070
Mean rank	**1.575**	2.175	2.900	4.225	4.125

Table 9: Same-declared-depth predictive scores at depth 5. Bold and underline mark first and second place, respectively.

Dataset	CAST	NODE	RADDT	GradTree	SGT-T
Classification: Accuracy					
Adult	**0.8635±0.0017**	0.8615±0.0020	0.8574±0.0020	0.8305±0.0068	0.8625±0.0013
Australian	**0.8551±0.0383**	0.8478±0.0441	0.8116±0.0644	0.8478±0.0316	0.8164±0.0084
Balance Scale	**0.9600±0.0277**	0.9520±0.0400	0.9280±0.0320	0.7440±0.0320	0.7920±0.0240
Bank Marketing	0.7878±0.0100	**0.7880±0.0072**	0.7829±0.0047	0.7651±0.0136	0.7862±0.0092
Banknote	**0.9976±0.0042**	0.9915±0.0076	**0.9976±0.0021**	0.9733±0.0128	0.9491±0.0167
Iris	0.9000±0.1000	0.9222±0.0385	**0.9333±0.0333**	**0.9333±0.0333**	**0.9333±0.0333**
Pendigits	0.9856±0.0014	**0.9883±0.0011**	0.9809±0.0020	0.7752±0.0250	0.8542±0.0105
Seismic Bumps	**0.9342±0.0000**	0.9304±0.0067	0.9052±0.0118	**0.9342±0.0000**	0.9310±0.0030
Spambase	**0.9385±0.0035**	0.9349±0.0065	0.9135±0.0140	0.8491±0.0268	0.8983±0.0023
Wine	**0.9722±0.0278**	0.9537±0.0321	0.8981±0.0578	0.8889±0.0481	0.8889±0.0556
Wisconsin Breast Cancer	**0.9643±0.0071**	0.9548±0.0165	0.9500±0.0143	0.9214±0.0311	0.9333±0.0109
Regression: R^2					
Abalone	**0.5680±0.0113**	0.5426±0.0154	0.4668±0.0255	0.4758±0.0166	0.4669±0.0373
Ailerons	**0.8431±0.0084**	0.8216±0.0050	0.7710±0.0098	0.6421±0.0050	0.6997±0.0091
California Housing	**0.7871±0.0092**	0.7721±0.0045	0.7313±0.0048	0.5053±0.0291	0.6742±0.0053
Concrete	**0.8744±0.0207**	0.8668±0.0161	0.7956±0.0512	0.4414±0.0521	0.7050±0.0346
CPU Act	**0.9812±0.0028**	0.9799±0.0038	0.9697±0.0057	0.9371±0.0079	0.9550±0.0066
Diabetes	0.4700±0.0247	**0.4736±0.0677**	0.1295±0.1387	0.3459±0.0806	0.1571±0.0698
Elevators	**0.9105±0.0022**	0.8926±0.0062	0.8927±0.0073	0.4788±0.0379	0.5596±0.0274
Friedman	**0.9586±0.0009**	0.9409±0.0009	0.8900±0.0012	0.5793±0.0223	0.7181±0.0036
Miami Housing 2016	**0.9017±0.0059**	0.8936±0.0014	0.8401±0.0091	0.6431±0.0054	0.7518±0.0144
Mean rank	**1.400**	2.225	3.325	4.275	3.775

Table 11: Feature support of retained internal nodes in deployed CAST trees. Each starting depth covers 20 datasets and three seeds. K counts exactly nonzero deterministic node–feature gates. Means and medians pool retained nodes, not trees. The final column gives the count and percentage of retained nodes with $K \leq 2$.

Starting depth	Retained nodes	Mean K	Median K	$K \leq 2$
2	170	14.49	8	8 (4.7%)
3	381	8.75	7	27 (7.1%)
4	762	7.71	7	76 (10.0%)
5	1,388	6.97	6	203 (14.6%)
6	2,633	6.99	6	477 (18.1%)

Table 10: CAST actual internal nodes (three-seed mean) by starting depth. Parentheses show the percentage of the corresponding complete tree's internal nodes retained. The reduction is 100% minus this percentage. Complete-tree counts are 3, 7, 15, 31, and 63.

Dataset	d=2	d=3	d=4	d=5	d=6
Classification					
Adult	3.00 (100.0%)	6.33 (90.5%)	11.00 (73.3%)	17.00 (54.8%)	24.00 (38.1%)
Australian	3.00 (100.0%)	6.67 (95.2%)	14.00 (93.3%)	23.67 (76.3%)	63.00 (100.0%)
Balance Scale	3.00 (100.0%)	7.00 (100.0%)	15.00 (100.0%)	30.00 (96.8%)	56.33 (89.4%)
Bank Marketing	3.00 (100.0%)	6.67 (95.2%)	12.67 (84.4%)	23.33 (75.3%)	41.00 (65.1%)
Banknote	3.00 (100.0%)	5.67 (81.0%)	10.67 (71.1%)	20.00 (64.5%)	40.33 (64.0%)
Iris	3.00 (100.0%)	6.33 (90.5%)	14.33 (95.6%)	22.33 (72.0%)	54.33 (86.2%)
Pendigits	3.00 (100.0%)	7.00 (100.0%)	15.00 (100.0%)	28.33 (91.4%)	46.67 (74.1%)
Seismic Bumps	0.33 (11.1%)	2.33 (33.3%)	3.67 (24.4%)	0.33 (1.1%)	0.33 (0.5%)
Spambase	3.00 (100.0%)	6.00 (85.7%)	9.67 (64.4%)	16.00 (51.6%)	27.33 (43.4%)
Wine	3.00 (100.0%)	6.00 (85.7%)	12.00 (80.0%)	16.00 (51.6%)	27.67 (43.9%)
Wisconsin Breast Cancer	2.33 (77.8%)	5.00 (71.4%)	8.33 (55.6%)	18.67 (60.2%)	32.33 (51.3%)
Regression					
Abalone	3.00 (100.0%)	7.00 (100.0%)	13.67 (91.1%)	25.67 (82.8%)	46.67 (74.1%)
Ailerons	3.00 (100.0%)	7.00 (100.0%)	14.67 (97.8%)	28.00 (90.3%)	51.33 (81.5%)
California Housing	3.00 (100.0%)	7.00 (100.0%)	15.00 (100.0%)	29.67 (95.7%)	59.00 (93.7%)
Concrete	3.00 (100.0%)	7.00 (100.0%)	15.00 (100.0%)	28.33 (91.4%)	53.67 (85.2%)
CPU Act	3.00 (100.0%)	6.00 (85.7%)	11.00 (73.3%)	20.33 (65.6%)	34.33 (54.5%)
Diabetes	3.00 (100.0%)	7.00 (100.0%)	14.00 (93.3%)	27.67 (89.2%)	54.67 (86.8%)
Elevators	3.00 (100.0%)	7.00 (100.0%)	14.33 (95.6%)	26.33 (84.9%)	53.00 (84.1%)
Friedman	3.00 (100.0%)	7.00 (100.0%)	15.00 (100.0%)	30.67 (98.9%)	53.33 (84.7%)
Miami Housing 2016	3.00 (100.0%)	7.00 (100.0%)	15.00 (100.0%)	30.33 (97.8%)	58.33 (92.6%)

A.2.2 Feature support in deployed trees

For each internal node retained after folding, we define K as the number of model-input features with a nonzero deterministic gate. Of the 300 fitted trees at starting depths two through six, nine fold to a constant root. The remaining trees retain 5,334 internal nodes. Table 11 summarizes their support sizes. In the primary configuration, 477 of 2,633 retained nodes (18.1%) use at most two model-input features. The mean and median K are 6.99 and 6. These counts refer to prepared model-input columns, whose number varies by dataset.

For these 2,633 nodes, the disjoint support-size bins contain 24 nodes with $K = 0$, 253 with $K = 1$, 200 with $K = 2$, 1,123 with $3 \leq K \leq 7$, 929 with $8 \leq K \leq 15$, and 104 with $K \geq 16$. A retained node can have $K = 0$ when its descendants still make the subtree input-dependent, so the node does not satisfy the constant-subtree folding condition. Constant-root trees contribute no internal nodes to this distribution.

Table 12: Depth-six ablations after 300 training epochs. Differences are variant minus CAST at the same stage, averaged over dataset-level three-seed means. W/T/L counts datasets where the variant is better, tied, or worse. Classification accuracy and regression R^2 are reported separately.

Variant	Δ Accuracy	W/T/L	ΔR^2	W/T/L
No CART transfer	-0.0023	2/4/5	-0.0134	2/0/7
Frozen basis	-0.0069	1/2/8	+0.0044	7/0/2
Fixed RBF basis	-0.0072	2/0/9	-0.0059	4/0/5
RBF-initialized basis	-0.0023	5/2/4	-0.0014	4/0/5
Fixed cubic B-spline basis	-0.0018	4/1/6	+0.0033	6/0/3
Early STG	-0.0015	3/2/6	-0.0027	6/0/3
Basis width 32	+0.0037	6/1/4	-0.0038	2/0/7
Basis width 128	-0.0002	3/1/7	-0.0035	3/0/6

Table 13: Node-response ablation at depth six after 300 epochs under a shared initialization with CART transfer disabled. Differences are relative to the response combining linear and nonlinear terms, averaged separately over classification accuracy and regression R^2 datasets.

Node response	Δ Accuracy	ΔR^2
Linear + nonlinear	0.0000	0.0000
Nonlinear only	+0.0019	-0.0060
Linear only	-0.0286	-0.0164

A.3 Expanded Ablations

The ablations use the same 20 datasets, seeds, and splits as the main comparison. Each variant is compared with CAST at the same evaluation stage. The 300-epoch linear-node sweep covers depths two through six. All other variants in Tables 12–14 cover 20 datasets and three seeds in the primary configuration. The no-transfer three-arm comparison uses the complete no-transfer node as its common reference, separating the linear-anchor effect from the trusted CART initialization. Removing the linear anchor from that node reduces mean regression R^2 by 0.0060, while its classification effect is small and positive. Freezing the shared basis lowers classification accuracy on average but improves regression R^2 on seven of nine datasets. Its batch-normalization buffers still update. The fixed cubic B-spline basis is included in the 300-epoch comparison in Table 12. Relative to CAST on the same dataset and seed after 300 epochs, it changes mean classification accuracy by -0.0018 and mean regression R^2 by $+0.0033$. This comparison does not include stochastic-gate continuation or exact folding. The early-STG variant has mean node support $K = 1.677$ across its 60 runs after continuation, with lower final mean accuracy and R^2 than CAST as shown in Table 14. The dense-continuation arm selects the original checkpoint after 300 epochs in 47 of 60 runs and a continuation checkpoint in 13. Its final comparison therefore measures the selected outcomes of the two pipelines rather than a pure effect of 100 additional epochs. Across all ablation variants, folding preserves the selected models' predictions to numerical precision.

A.3.1 Training time and resource observations

Table 15 reports recorded training time and sampled memory for each ablation arm. Training time is available for all 60 dataset–seed runs per arm. Memory sampling is incomplete for some arms during the first 300 epochs, and sampled GPU maxima may fall below the true per-run peaks. Folding is a deterministic transformation and has no training-time record.

Table 14: Depth-six ablations after the complete gate-and-fold pipeline. Differences are variant minus CAST at the final prediction endpoint, averaged over dataset-level three-seed means. W/T/L is from the variant's perspective. The dense continuation arm starts from the same 300-epoch checkpoint but uses no stochastic gates.

Variant	Δ Accuracy	W/T/L	ΔR^2	W/T/L
Linear node	-0.0346	0/1/10	-0.0156	3/0/6
No CART transfer	-0.0030	3/4/4	-0.0129	3/0/6
Dense continuation	-0.0059	2/3/6	-0.0015	2/0/7
Early STG	-0.0035	1/3/7	-0.0038	6/0/3
Basis width 32	-0.0030	4/2/5	-0.0037	4/0/5
Basis width 128	-0.0039	2/3/6	-0.0029	3/0/6
STG penalty $0.003/63$	-0.0061	2/7/2	-0.0004	4/0/5
STG penalty $0.03/63$	+0.0003	2/5/4	-0.0003	2/0/7

Table 15: Recorded training time and sampled resource use for depth-six ablations. Each row covers 60 dataset–seed runs. The stages are the first 300 training epochs and the following 100 continuation epochs. Time is the median recorded training time per run in seconds. Memory entries give the median and the number of runs sampled. VM HWM is process high-water memory in MiB. GPU used memory is the sampled maximum in MB, which can underestimate the true peak. A dash denotes no samples.

Stage	Variant	Time (s)	VM HWM (MiB)	GPU used (MB)
Initial 300	Linear node	186.2	– (0/60)	– (0/60)
Initial 300	Nonlinear only	264.6	1700 (60/60)	536 (60/60)
Initial 300	Linear only	224.3	1695 (60/60)	536 (60/60)
Initial 300	No CART transfer	278.6	1708 (40/60)	544 (40/60)
Initial 300	Frozen basis	242.9	1696 (60/60)	536 (60/60)
Initial 300	Fixed RBF basis	272.3	1676 (60/60)	536 (60/60)
Initial 300	RBF-initialized basis	296.6	1717 (60/60)	536 (60/60)
Initial 300	Early STG	314.7	1750 (60/60)	536 (60/60)
Initial 300	Basis width 32	299.6	1711 (60/60)	535 (60/60)
Initial 300	Basis width 128	257.7	1705 (60/60)	534 (60/60)
Next 100	Linear node	83.4	1650 (60/60)	494 (60/60)
Next 100	No CART transfer	97.8	1654 (60/60)	496 (60/60)
Next 100	Dense continuation	88.9	1563 (60/60)	496 (60/60)
Next 100	Early STG	100.2	1653 (60/60)	494 (60/60)
Next 100	Basis width 32	110.2	1653 (60/60)	495 (60/60)
Next 100	Basis width 128	105.9	1656 (60/60)	497 (60/60)
Next 100	STG penalty $0.003/63$	98.4	1652 (60/60)	495 (60/60)
Next 100	STG penalty $0.03/63$	105.7	1653 (60/60)	496 (60/60)