NADT: NEIGHBORHOOD-AUGMENTED NEURAL DECISION TREES

Anonymous authors
Paper under double-blind review

ABSTRACT

Neural decision trees enable end-to-end gradient optimization of hierarchical routing. Their split functions are globally fixed during inference, applying identical decision boundaries irrespective of local data structure. Meanwhile, k-nearest neighbor methods provide sample-adaptive local awareness, but their flat architectures cannot decompose complex boundaries hierarchically. To address this limitation, we propose NADT, which equips a neural decision tree with neighborhood-adaptive routing through a Neighborhood Context Module (NCM) and a soft decision tree ensemble. NCM maintains M learnable prototypes in a shared embedding space and maps sample-to-prototype distances via temperature-scaled Softmax to node-level biases. These biases are injected into the oblique split nodes of the soft tree ensemble, shifting decision boundaries dynamically with each sample's neighborhood context. Since prototypes are learned parameters, NADT achieves local awareness without retrieving the training set at inference. Extensive experiments on 30 public datasets show that NADT attains the best average rank among 11 competing methods, with ablation confirming module effectiveness and visualization rendering the decision process inspectable. The source code for NADT will be publicly available[1].

1 INTRODUCTION

Decision trees and their ensembles are among the most effective methods for many domains (Grinsztajn et al., 2022; Shwartz-Ziv & Armon, 2022). Neural decision trees further advance this paradigm through differentiable oblique splits and soft routing (Kontschieder et al., 2015; Popov et al., 2020), enabling end-to-end gradient optimization. However, their split functions remain globally fixed at inference, applying identical decision boundaries to all samples regardless of local data structure. Real-world data often exhibits local heterogeneity, where different regions of the feature space require different optimal decision boundary directions. Figure 1 illustrates this phenomenon on a synthetic dataset with three regions, each having a distinct optimal split direction separated by 120°. As shown in the figure, DTSemNet (Panda et al., 2026) produces distorted polygonal boundaries at region interfaces where a single global split cannot accommodate varying local directions.

Recent deep k-nearest neighbor methods provide a natural way to introduce such local information into tabular prediction (Nader et al., 2022). Through instance retrieval and metric learning, they adapt predictions according to the neighborhood of each query. However, their prediction process is typically flat and lacks the hierarchical decomposition of decision trees. Moreover, their globally shared retrieval space or distance metric may mix samples from regions with different local separation patterns, especially near region interfaces. As shown in Figure 1, this can lead to irregular or biased boundaries for TabR (Gorishniy et al., 2024) and ModernNCA (Ye et al., 2025).

Taken together, the comparisons in Figure 1 suggest that neither fixed hierarchical routing nor flat neighborhood modeling alone is sufficient. What is needed is a model that uses local neighborhood information to adapt the routing decisions of a tree, while preserving its hierarchical decision structure (Borisov et al., 2024; Somvanshi et al., 2026). To this end, we propose the Neighborhood-Augmented Neural Decision Tree (NADT), which introduces neighborhood-conditioned routing into a neural decision tree through a Neighborhood Context Module (NCM).

[1] https://anonymous.4open.science/r/NADT-C314

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.

Specifically, NCM maintains M learnable prototypes as compact representatives of latent neighborhoods in a shared embedding space. For each input, its distances to the prototypes are converted into affinity weights using a temperature-scaled Softmax, producing a neighborhood representation specific to that input. This representation is then mapped to node-level routing biases $\boldsymbol{\delta}$, which are added to the oblique split activations. Consequently, the same internal node can adapt its routing behavior to samples from different neighborhoods, rather than applying an identical split rule to every sample reaching that node. Notably, since the prototypes are learned model parameters, NADT captures neighborhood information without storing or retrieving training samples at inference.

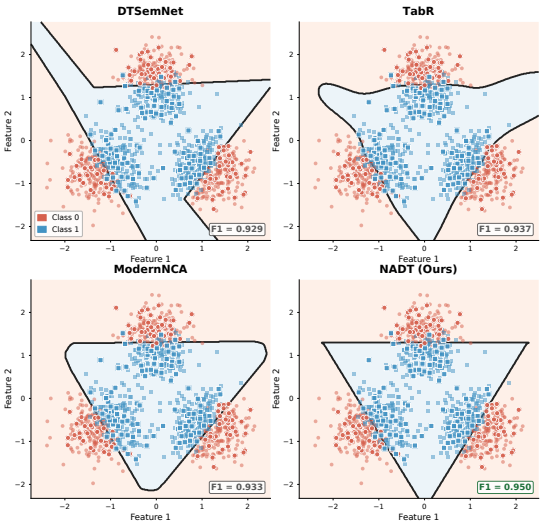


Figure 1: Decision boundaries on a locally heterogeneous synthetic dataset. Three regions have optimal split directions separated by 120°. DTSemNet, TabR, and ModernNCA produce distorted boundaries at region interfaces. NADT adapts to each region's local geometry, producing clean and accurate boundaries.

As shown in Figure 1, NADT produces coherent boundary segments that adapt to the distinct separation patterns of the three regions, providing an intuitive illustration of its neighborhood-conditioned routing. Beyond this synthetic example, extensive experiments demonstrate NADT's strong predictive performance, with NADT achieving the best average rank among tree-based and nearest-neighbor baselines. Further ablation and visualization studies verify the effectiveness of the proposed components and reveal how the learned neighborhood representations adapt the decision. Our main contributions are:

- We propose NADT, a framework that integrates neighborhood awareness into hierarchical routing via a feature projector, a Neighborhood Context Module (NCM), and a soft decision tree ensemble, jointly trained end-to-end.
- We design NCM, which uses M learnable prototypes and temperature-scaled Softmax to map neighborhood context to sample-adaptive node-level biases, achieving local awareness without training-set retrieval.
- We conduct systematic comparisons on public datasets, where NADT achieves the best overall performance and is competitive in both effectiveness and efficiency.

2 Related Work

2.1 Neural Decision Trees

Decision trees decompose complex feature spaces into interpretable decision rules, making them among the most effective inductive biases for tabular data (Hyafil & Rivest, 1976; Murthy et al., 1994). Recent work also studies the sample/feature trade-off in reproducible evaluation of tree ensembles (Pandita et al., 2026). Making decision trees differentiable enables end-to-end gradient optimization while preserving hierarchical structure. Foundational work includes Deep Neural Decision Forests (Kontschieder et al., 2015) and implicit oblique tree representations via ReLU networks (Lee & Jaakkola, 2020). NODE (Popov et al., 2020) and NCART (Luo & Xu, 2024) build oblivious tree ensembles as neural layers. For oblique splits, DTSemNet (Panda et al., 2026) encodes hard trees as semantically equivalent ReLU networks Hinge regression trees (Li et al., 2026) learn splits via Newton optimization. DOFEN (Chen et al., 2024) achieves on-off sparse column selection. GRANDE (Marton et al., 2024b) and GradTree (Marton et al., 2024a) learn axis-aligned trees via gradient descent, and RADDT (Mao et al., 2025) reformulates hard splits for exact differentiable optimization. SoHoT (Köbschall et al., 2025) extends differentiable trees to data streams.

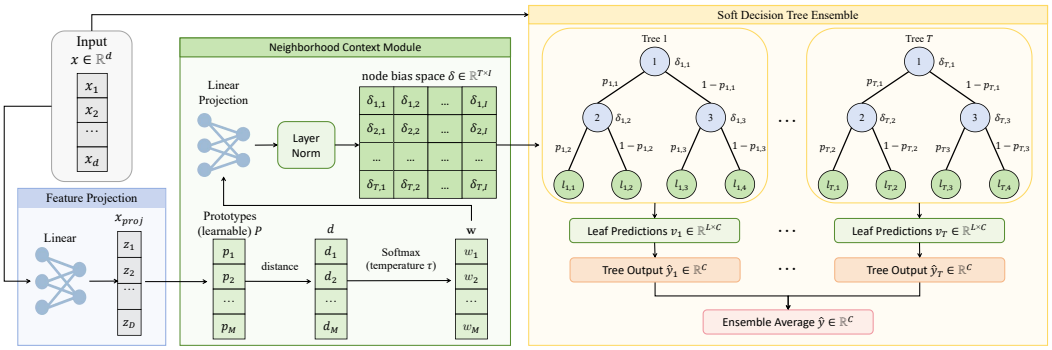


Figure 2: Overview of the NADT framework. The feature projector maps raw features to an embedding space. The Neighborhood Context Module (NCM) generates sample-adaptive node biases $\boldsymbol{\delta}$ from learnable prototypes. The soft decision tree ensemble uses these biases for personalized routing and aggregates leaf predictions.

Soft decision trees have been applied to survival analysis (Consolo et al., 2026). Despite these advances, existing neural trees do not consider local context at nodes, limiting adaptivity to feature spaces. NADT addresses this by injecting prototype-derived biases into the split activation.

2.2 Deep Neighborhood Methods

The k-nearest neighbor paradigm provides sample-adaptive predictions by retrieving similar training instances. Complementarily, prototype-based methods learn compact representations of local distributions. Prototypical Networks (Snell et al., 2017) classify images by computing distances to learned class prototypes. ProtoTree (Nauta et al., 2021) places trainable prototypical parts at tree nodes for image recognition, combining prototype learning with hierarchical routing. TabR (Gorishniy et al., 2024) embeds KNN retrieval within a feed-forward network. ModernNCA (Ye et al., 2025) revisits Neighborhood Components Analysis with deep representations. ProtoGate (Jiang et al., 2024) uses prototype-based prediction for high-dimensional biomedical data. DNNR (Nader et al., 2022) improves KNN regression via local gradient estimation. Some works have adapted these ideas to in-context prediction and medical data (Guta, 2026; Karolczak & Stefanowski, 2025). In the broader landscape, attention-based architectures (Arik & Pfister, 2021; Gorishniy et al., 2022; Leng et al., 2026) and parameter-efficient ensembles (Wen et al., 2020; Gorishniy et al., 2025) achieve better performance through globally shared computations. In summary, deep neighborhood methods capture local structure but rely on globally shared retrieval spaces, which can distort decision boundaries near region interfaces (Borisov et al., 2024; Somvanshi et al., 2026; Li et al., 2025). NADT addresses this by using a shared prototype bank to generate sample-adaptive routing biases.

3 Method

3.1 Problem Formulation and Overview

Let $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$ be a dataset, where $\mathbf{x}_i \in \mathbb{R}^d$ is a d-dimensional feature vector and $y_i \in \{0, 1, \ldots, C-1\}$ is the class label. The goal is to learn a classifier $f : \mathbb{R}^d \to \mathbb{R}^C$ that minimizes prediction error on unseen samples. We propose a Neighborhood-Augmented Neural Decision Trees (NADT), as shown in Figure 2. NADT consists of three jointly trained modules. Given an input $\mathbf{x} \in \mathbb{R}^d$, the overall framework can be expressed as:

$$f(\mathbf{x}) = \frac{1}{T} \sum_{t=1}^{T} \sum_{\ell=1}^{L} \mu_{t,\ell}\Big(\mathbf{x}_{\text{sub}}^{(t)},\ \boldsymbol{\delta}(\mathbf{x}_{\text{proj}})\Big) \cdot \mathbf{v}_{t,\ell} \tag{1}$$

where T is the number of trees and $L = 2^{\text{depth}}$ is the number of leaf nodes per tree. Here, $\mathbf{x}_{\text{sub}}^{(t)}$ is the feature subset input for the t-th tree, $\mathbf{x}_{\text{proj}}$ is the projected embedding, $\boldsymbol{\delta}(\mathbf{x}_{\text{proj}})$ denotes the sample-adaptive node biases generated by NCM, $\mu_{t,\ell}$ denotes the leaf reach probability, and $\mathbf{v}_{t,\ell}$ is the learnable prediction vector at each leaf.

3.2 NEIGHBORHOOD CONTEXT MODULE

Input features $\mathbf{x} \in \mathbb{R}^d$ are first mapped to an embedding space through a shared linear projection layer:

$$\mathbf{x}_{\text{proj}} = \mathbf{W}_{\text{proj}}\mathbf{x} + \mathbf{b}_{\text{proj}} \in \mathbb{R}^D \tag{2}$$

NCM maintains M learnable prototype vectors $\{\mathbf{p}_m\}_{m=1}^M$ in the embedding space $\mathbb{R}^D$, serving as a parametric representation of local data distributions. To accelerate convergence, prototypes are initialized via Mini-Batch K-means clustering of the projected training embeddings, whose cluster centers serve as initial prototype values.

Based on these prototypes, given a projected embedding $\mathbf{x}_{\text{proj}}$, the squared Euclidean distance to each prototype is computed:

$$d_m = \|\mathbf{x}_{\text{proj}} - \mathbf{p}_m\|_2^2, \quad m = 1, \ldots, M \tag{3}$$

where M is the number of prototypes. The distances are then converted into affinity weights using a Softmax function with a learnable temperature τ, parameterized in log space to ensure positivity:

$$w_m = \frac{\exp(-d_m/\tau)}{\sum_{m'=1}^M \exp(-d_{m'}/\tau)}, \quad \mathbf{w} \in \mathbb{R}^M \tag{4}$$

Temperature τ controls the sharpness of neighborhood perception. A smaller τ concentrates weights on the nearest prototype, while a larger τ makes the weights more uniform. $\mathbf{w}$ can be viewed as the soft nearest-neighbor assignment of the current sample in the prototype space. This soft partition defines a *neighborhood* per prototype, with w_m quantifying the current sample's membership in the m-th one.

The affinity weights $\mathbf{w}$ are subsequently mapped to the node bias space via a linear projection layer, with LayerNorm to stabilize magnitude fluctuations in early training:

$$\boldsymbol{\delta} = \text{LayerNorm}(\mathbf{W}_{\text{ctx}}\mathbf{w}) \in \mathbb{R}^{T \times I} \tag{5}$$

where $I = 2^{\text{depth}} - 1$ is the number of internal nodes per tree, and $\mathbf{W}_{\text{ctx}} \in \mathbb{R}^{(T \cdot I) \times M}$ is a learnable weight matrix. Each dimension (t, i) of $\boldsymbol{\delta}$ corresponds to the personalized bias correction for the i-th internal node of the t-th tree. This bias is then injected into the split activation function of the soft decision tree, allowing the same split boundary to shift dynamically according to the sample's neighborhood context.

3.3 SOFT DECISION TREE ENSEMBLE

NADT constructs T soft neural decision trees, each of depth, using oblique splits for feature partitioning. To enhance ensemble diversity, each tree randomly samples $F_{\text{sub}} = \lfloor d \times r \rfloor$ feature indices at initialization, where r is the feature sampling ratio.

At internal node i of the t-th tree, the split activation consists of a linear combination of the feature subset plus the bias correction from NCM:

$$r_{t,i} = \sum_{f=1}^{F_{\text{sub}}} \mathbf{x}_{\text{sub},f}^{(t)} \cdot w_{t,i,f}^{\text{split}} + b_{t,i}^{\text{split}} + \delta_{t,i} \tag{6}$$

where $w_{t,i,f}^{\text{split}}$ and $b_{t,i}^{\text{split}}$ are learnable split weights and bias. $\delta_{t,i}$ is the sample-adaptive bias from NCM that enables the split boundary to shift dynamically with neighborhood context. $\mathbf{w}^{\text{split}}$ determines which features matter. For a *fixed* neighborhood context, $\delta_{t,i}$ acts as a scalar offset that translates the boundary along its normal while leaving its orientation unchanged; because $\delta_{t,i}$ itself varies with the input's neighborhood, however, the effective decision surface is nonlinear and region-dependent, so the same node can place its boundary differently in different regions. Since $\partial p_{t,i}/\partial \delta_{t,i} = 1/(2(1 + |r_{t,i}|)^2)$, the bias has the strongest influence when a sample lies near the boundary.

Its effect diminishes as the sample moves away, naturally prioritizing adaptive routing where decisions are most uncertain. The split activation is then smoothed via the softsign function and mapped to the $(0, 1)$ interval as the probability of routing to the right child:

$$p_{t,i} = \frac{\text{softsign}(r_{t,i}) + 1}{2} = \frac{r_{t,i}/(1 + |r_{t,i}|) + 1}{2} \tag{7}$$

Based on the above routing probabilities, the reach probability of leaf node ℓ equals the product of routing probabilities along the path from the root to that leaf. If the path takes the left branch at node i, it multiplies by $1 - p_{t,i}$; if it takes the right branch, it multiplies by $p_{t,i}$:

$$\mu_{t,\ell} = \prod_{(i,\mathrm{dir}) \in \mathrm{path}(\ell)} \begin{cases} p_{t,i} & \mathrm{dir} = \mathrm{right} \\ 1 - p_{t,i} & \mathrm{dir} = \mathrm{left} \end{cases} \tag{8}$$

Each leaf node ℓ holds a learnable prediction vector $\mathbf{v}_{t,\ell} \in \mathbb{R}^C$, interpreted as class logits. The output of a single tree is the expectation of leaf predictions with respect to reach probabilities: $\hat{\mathbf{y}}_t = \sum_{\ell=1}^{L} \mu_{t,\ell} \cdot \mathbf{v}_{t,\ell}$ The final prediction is the simple average over all T trees: $\hat{\mathbf{y}} = \frac{1}{T} \sum_{t=1}^{T} \hat{\mathbf{y}}_t$, and class probabilities are obtained by applying Softmax to $\hat{\mathbf{y}}$.

3.4 Training Objective

Given the forward computation described above, the total training loss of NADT consists of a classification loss and a prototype diversity regularization term:

$$\mathcal{L} = \mathcal{L}_{\text{CE}} + \lambda_{\text{div}} \cdot \mathcal{L}_{\text{div}} \tag{9}$$

where $\mathcal{L}_{\text{CE}}$ is the multi-class cross-entropy. A diversity regularization term prevents prototype collapse, where all prototypes would converge to the same location and lose their distribution-representation capacity. It penalizes prototypes whose pairwise distance falls below a margin m, using a hinge loss:

$$\mathcal{L}_{\text{div}} = \mathbb{E}_{a \neq b} \left[\max(0, m - \|\mathbf{p}_a - \mathbf{p}_b\|_2) \right] \tag{10}$$

This loss is zero for pairs whose distance exceeds m, only pushing apart prototypes that are too close. Since $\mathcal{L}_{\text{div}} \in [0, m]$, the training objective is bounded without additional constraints. where λ_{div} is the regularization coefficient and m is the desired minimum pairwise distance.

The training algorithm is provided in Appendix A, and implementation details are given in Appendix G.

3.5 Theoretical Analysis

Expressive power of NCM biases. Fixing the architecture (the number of trees T, their depths, and per-node feature subsets), let $\mathcal{F}_{\text{base}}$ be the function class of the soft tree ensemble without NCM, and $\mathcal{F}_{\text{NADT}}$ the class with NCM-generated biases.

Theorem 1. *$\mathcal{F}_{base} \subsetneq \mathcal{F}_{NADT}$ whenever the prototype configuration is nondegenerate, i.e., $M \geq 2$ with distinct prototypes and $\mathbf{W}_{ctx} \neq \mathbf{0}$.*

The inclusion holds since setting $\mathbf{W}_{\text{ctx}} = \mathbf{0}$ recovers the baseline. For strictness, under this condition the affinity vectors $\mathbf{w}(\mathbf{x})$ are non-constant over $\mathbf{x}$, so the resulting biases produce routing probabilities unattainable under any fixed-split configuration. This provides a theoretical basis for the boundary adaptation in Figure 1.

Temperature-controlled localization. The granularity of neighborhood perception in NCM is governed by the temperature τ.

Lemma 2. *As $\tau \to 0^+$, the affinity weights $\mathbf{w}(\mathbf{x})$ converge to hard nearest-neighbor assignment over the prototypes, making $\boldsymbol{\delta}(\mathbf{x})$ piecewise-constant over the Voronoi cells; as $\tau \to \infty$, $\mathbf{w}(\mathbf{x}) \to 1/M$ and $\boldsymbol{\delta}$ degenerates to a constant.*

Hence τ interpolates between hard nearest-neighbor routing and a constant global bias, so the effective neighborhood scale is fully controlled by τ, which is learned end-to-end and adapts to each dataset.

Gradient stability of softsign routing. In soft decision trees, gradients propagate through a chain of routing functions along the root-to-leaf path.

Theorem 3. *For any internal node on the path to leaf ℓ at depth D, the log-leaf-reach-probability gradient satisfies $|\partial \log \mu_{t,\ell} / \partial r_{t,i_k}| \geq 1/(2(1+R)^2)$, where R bounds $|r_{t,i}|$. This lower bound is independent of D.*

The bound follows from the softsign derivative $\partial p_{t,i}/\partial r_{t,i} = 1/(2(1 + |r_{t,i}|)^2)$ and the additive decomposition of $\log \mu_{t,\ell}$ over path nodes, which keeps the per-node gradient lower bound independent of depth D. Under sigmoid routing, the same log-path gradient is bounded below by $\sigma(-R)$, which decays *exponentially* in the margin R, whereas the softsign bound $1/(2(1 + R)^2)$ decays only *polynomially*. Softsign therefore retains informative gradients for confident samples far from decision boundaries, enabling effective end-to-end training without Straight-Through estimators.

Furthermore, since $\partial p_{t,i}/\partial \delta_{t,i}$ shares the same form, NCM biases receive the strongest gradient near decision boundaries ($|r_{t,i}| \approx 0$) and negligible gradient for confident samples, prioritizing boundary-uncertain routing. Complete proofs and auxiliary lemmas are provided in Appendix B.

4 Experiments

4.1 Experimental Setup

All experiments use public datasets from OpenML (Vanschoren et al., 2014)[2]. We select 30 datasets covering both binary and multi-class tasks, with sample sizes ranging from 60 to 99,289, feature dimensions spanning from 4 to 54,621, class counts from 2 to 11. All input features are preprocessed with Min-Max normalization, scaling each feature to the $[0, 1]$ range.

We use F1(Macro) as the primary evaluation metric to comprehensively assess model performance across all classes. All models undergo 5 repetitions of 5-fold cross-validation on each dataset, yielding results for averaging. We employ the Friedman test (significance level $\alpha = 0.05$) to assess whether statistically significant differences exist among models, followed by the Nemenyi post-hoc test for pairwise comparisons. Complete dataset statistics and standard deviations are provided in Appendix D, ablation results in Appendix F, hyperparameter sensitivity in Appendix H, and runtime analyses in Appendix I.

4.2 Comparative Experiments

We compare NADT against 11 baseline methods. The traditional group includes RandomForest (Breiman, 2001), XGBoost (Chen & Guestrin, 2016), LightGBM (Ke et al., 2017), and CatBoost (Prokhorenkova et al., 2018). The deep learning group includes TabR (Gorishniy et al., 2024), GRANDE (Marton et al., 2024b), DOFEN (Chen et al., 2024), ModernNCA (Ye et al., 2025), ProtoGate (Jiang et al., 2024), SoHoT (Köbschall et al., 2025), and DTSemNet (Panda et al., 2026). Hyperparameters are selected via Optuna (Akiba et al., 2019) using 25% of the training set as a validation split, and the best setting is used to train the model for testing. Detailed method descriptions and hyperparameter search spaces are provided in Appendix E.

Table 1 reports the F1(Macro) scores of all methods across 30 datasets. NADT achieves the highest score on 14 datasets and the second-highest on 6, demonstrating strong performance across diverse conditions. Each of the traditional tree ensembles only achieves the highest score on at most two datasets. TabR achieves the highest score on 6 datasets through instance retrieval, and ModernNCA on 4 datasets through class-center distances. Both provide local awareness but lack hierarchical routing, and do not consistently outperform NADT on datasets requiring hierarchical feature decomposition. Neural decision tree methods (GRANDE, DTSemNet, SoHoT, DOFEN) retain hierarchical routing but share the same limitation of globally fixed splits. GRANDE achieves the top score only on OVA_Breast, with instance-wise weighting operating at the prediction level rather than the routing level. DOFEN and SoHoT exhibit instability on specific tasks (thyroid: 0.274, cortex_nuclear: 0.3933), and none introduces neighborhood awareness into split decisions. ProtoGate runs on 22 of 30 datasets, as its HDLSS-oriented design incurs GPU memory overflow on moderate to large datasets (denoted as "—"). Overall, NADT attains the best average rank by combining neighborhood-adaptive routing with hierarchical tree-structured decision making.

Figure 3 shows the Nemenyi post-hoc critical difference (CD) diagram (ProtoGate is excluded due to missing results on 8 datasets). The top tier (average ranks 2.27–3.63) consists of NADT, TabR, and ModernNCA. TabR and ModernNCA both leverage neighborhood retrieval, suggesting that neighborhood information is broadly effective for tabular classification. Traditional tree ensembles

[2] https://www.openml.org/

Table 1: F1(Macro) scores across 30 datasets. Best results are in bold; second-best are underlined. "—" indicates that the method could not run on the dataset.

Dataset	NADT	RF	XGBoost	LGBM	CatBoost	MNCA	TabR	P.Gate	DOFEN	GRANDE	SoHoT	DTSem
thyroid	0.9458	0.9369	0.9275	0.9297	0.9427	0.9435	**0.9462**	0.9009	0.2740	0.9198	0.6058	0.9285
cortex_nuclear	**0.9972**	0.9919	0.9724	0.9887	0.9875	0.9964	0.9938	0.9955	0.8394	0.9273	0.3933	0.9814
optdigits	0.9806	0.9769	0.9816	0.9841	0.9801	**0.9900**	0.9876	—	0.9776	0.9720	0.7088	0.9384
texture	0.9976	0.9728	0.9838	0.9869	0.9796	**0.9995**	0.9990	—	0.9888	0.9602	0.7355	0.9791
banknote-auth.	**1.0000**	0.9932	0.9962	0.9951	0.9966	0.9991	0.9991	0.9993	0.9967	0.9882	0.9742	0.9893
sonar	**0.8775**	0.8130	0.8433	0.8472	0.8437	0.8744	0.8601	0.8430	0.6703	0.8366	0.7641	0.8449
pollen	0.4918	0.4919	0.4975	**0.5016**	0.4939	0.4919	0.4600	—	0.3335	0.4966	0.4810	0.4844
elevators	0.8667	0.8270	0.8700	0.8673	0.8561	**0.8860**	0.8856	—	0.8544	0.8489	0.8580	0.8808
JapaneseVowels	0.9926	0.9775	0.9842	0.9875	0.9809	0.9980	**0.9982**	—	0.9668	0.9687	0.9342	0.9781
AP_Colon_Kidney	**0.9805**	0.9691	0.9709	0.9721	0.9717	0.9794	0.9761	0.9732	0.3458	0.9725	0.9761	0.9680
AP_Endo._Prostate	**0.9984**	0.9846	0.9892	0.9692	0.9831	0.9938	0.9892	0.9952	0.3628	0.9845	0.9376	0.9628
AP_Breast_Omentum	0.9368	0.9378	**0.9442**	0.9426	0.9316	0.9317	0.9416	0.9215	0.4497	0.9275	0.8154	0.8783
AP_Omentum_Kidney	0.9636	0.9611	0.9593	0.9621	0.9619	0.9508	**0.9640**	0.9577	0.4355	0.9560	0.8950	0.9194
AP_Omentum_Lung	0.9177	0.9193	0.9108	0.9184	**0.9239**	0.9174	0.9109	0.8633	0.3830	0.9128	0.9123	0.8613
AP_Uterus_Kidney	0.9717	0.9678	0.9698	**0.9732**	0.9683	0.9674	0.9684	0.9692	0.4037	0.9568	0.8893	0.9579
AP_Endo._Ovary	0.8869	0.8314	0.8921	0.8736	0.8832	0.8792	**0.9050**	0.8699	0.4333	0.8368	0.8110	0.7343
hiva_agnostic	**0.6871**	0.6342	0.6653	0.6256	0.6397	0.6530	0.6557	—	0.4910	0.5763	0.4910	0.6132
eating	**0.6468**	0.5235	0.5897	0.6011	0.5646	0.6369	0.6320	0.5251	0.1920	0.4908	0.1240	0.5043
tumors_C	**0.5752**	0.4636	0.5100	0.4802	0.4957	0.5057	0.5691	0.5553	0.3937	0.4327	0.3937	0.4641
ovarianTumour	**0.6979**	0.5262	0.6002	0.6346	0.4993	0.6534	0.6824	0.5935	0.3094	0.4677	0.3094	0.5588
BurkittLymphoma	0.8387	0.7473	0.7959	0.8177	0.7547	**0.8713**	0.8280	0.8527	0.2452	0.7221	0.2452	0.7242
OVA_Breast	0.9598	0.9479	0.9576	0.9572	0.9588	0.9460	0.9511	0.9524	0.4374	**0.9613**	0.9376	0.9230
OVA_Uterus	**0.7873**	0.6593	0.7801	0.7586	0.7422	0.7732	0.7660	0.6986	0.4791	0.7075	0.5208	0.7289
OVA_Kidney	0.9748	0.9699	0.9738	0.9745	**0.9771**	0.9714	0.9704	0.9729	0.4541	0.9694	0.9396	0.9507
OVA_Lung	**0.7873**	0.6492	0.7688	0.7657	0.7219	0.7669	0.7779	0.6818	0.4791	0.7075	0.5208	0.7289
OVA_Omentum	**0.7085**	0.4965	0.6832	0.6749	0.6050	0.6969	0.6932	0.6736	0.4872	0.5723	0.4873	0.6346
OVA_Endometrium	**0.7157**	0.4928	0.6476	0.6264	0.5663	0.6774	0.6717	0.5986	0.4899	0.5934	0.4899	0.6496
OVA_Prostate	**0.9885**	0.9806	0.9804	0.9805	0.9797	0.9756	0.9846	0.9816	0.4886	0.9733	0.9704	0.9219
jannis	0.5488	0.5187	0.5525	0.5623	0.5365	0.5303	**0.6061**	—	0.5214	0.4664	0.4921	0.4902
SVHN	0.8427	0.7080	0.7190	0.7515	0.7193	0.8335	**0.8815**	—	0.8194	0.4672	0.3038	0.2623

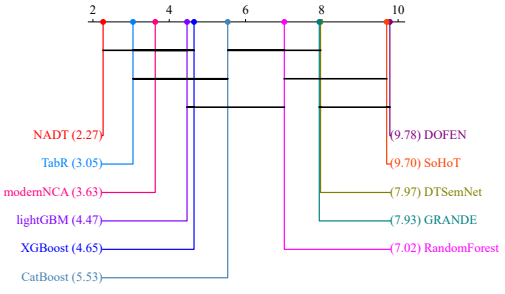


Figure 3: Critical difference diagram of the Nemenyi post-hoc test on F1(Macro) ($\alpha = 0.05$).

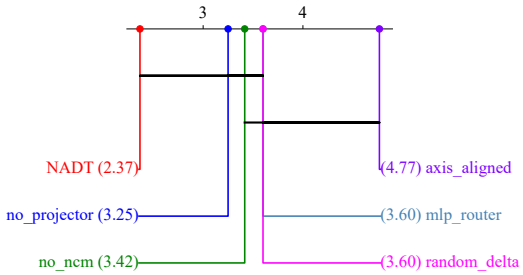


Figure 4: Critical difference diagram of the ablation study (Friedman test, $\alpha = 0.05$).

occupy the middle tier (4.47–7.02), while neural decision trees rank in the bottom tier (7.93–9.78). NADT is the only neural tree that reaches the top tier, indicating that neighborhood-adaptive routing mitigates the limitations of globally fixed splits.

4.3 Ablation Study

We conduct ablation studies on four key components of NADT, together with a capacity-matched control that replaces NCM with a plain MLP. All variants employing default hyperparameters and following the same evaluation protocol as the main experiments. Figure 4 shows the CD diagram of the Friedman test ($\alpha = 0.05$). The test statistic $\tau_F = 25.53$ far exceeds the critical value of 2.28, and full NADT (average rank 2.37) achieves the best rank among all variants, significantly outperforming the axis-aligned variant under the Nemenyi test (CD $= 1.38$). **w/o NCM** ($\boldsymbol{\delta} = \mathbf{0}$) reduces the model to a plain soft decision tree ensemble and causes noticeable degradation on most datasets, with sharp drops where samples are scarce or classes overlap. **w/ Random $\boldsymbol{\delta}$** replaces the prototype projection with a frozen random matrix and performs nearly identically to w/o NCM, confirming that NCM's gains stem from learned neighborhood structure rather than random perturbation. **w/ MLP Router** replaces NCM with a two-layer MLP of matched parameter count that maps the projected embedding directly to biases, and drops to the level of w/ Random $\boldsymbol{\delta}$ (average rank 3.60). This indicates that the gains stem from the prototype-based neighborhood structure itself rather than generic input-conditioned routing. **w/o Projector** computes prototype distances in the raw feature space, causing a small overall drop with more pronounced degradation on high-dimensional datasets.

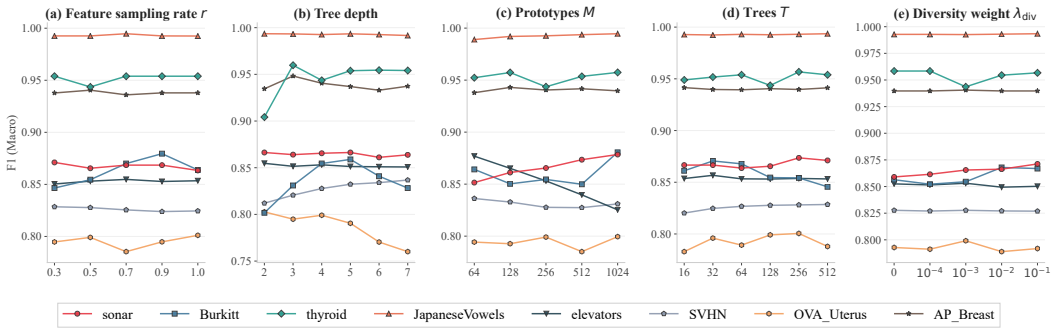


Figure 5: Hyperparameter sensitivity analysis on 8 representative datasets. Each subplot varies one hyperparameter while fixing others at their default values.

w/ Axis-Aligned replaces oblique splits with single-feature thresholds and suffers the most severe degradation, particularly on multi-class tasks requiring feature interactions. Overall, all components contribute to NADT's performance.

4.4 Hyperparameter Sensitivity

We evaluate NADT's sensitivity to key hyperparameters via univariate control experiments on 8 representative datasets (Figure 5). Overall, NADT exhibits robustness to hyperparameter variations, with performance remaining stable across a wide range of settings. Performance remains stable across feature sampling ratio r, with lower values increasing ensemble diversity on simple tasks and higher values benefiting small-sample tasks. **Tree depth** determines the expressive capacity of each soft decision tree. Shallow trees suffice for simple tasks but underfit on complex ones, while excessive depth causes degradation on some datasets due to the multiplicative decay of path probabilities, yet continues to improve performance on large-scale tasks. Too few prototypes fail to cover data diversity, while too many cause affinity weights to approach a uniform distribution, leading to sharp degradation on some datasets; the optimal number varies considerably across tasks. Ensemble performance improves rapidly from $T = 16$ to $T = 64$ and then saturates, with larger ensembles bringing more significant gains on complex tasks. **Diversity loss weight λ_{div}** prevents prototype collapse via a hinge loss. Removing the diversity loss λ_{div} causes clear degradation on some datasets, while a few perform better without it.

4.5 Visualization Analysis

To intuitively illustrate NADT's internal decision mechanism and prototype learning behavior, we conduct visualization analysis on the sonar dataset. Figure 6 visualizes NADT's model behavior. Figure 6(a) shows the top-10 features by accumulated absolute split weights across all trees. The weights are distributed uniformly with no single feature dominating, indicating that oblique splits effectively combine multiple features rather than relying on any single one. Figure 6(b) shows the absolute change in routing probability $|\Delta p_{t,i}|$ caused by δ, plotted against $|r_{t,i}^{(0)}|$, the activation magnitude without δ. The impact decreases monotonically from approximately 0.26 near the decision boundary to 0.026 far from it. This confirms that the additive bias exerts the strongest influence when the model is most uncertain, while leaving confident decisions largely undisturbed.

Figure 7 visualizes NADT's prototype behavior. Figure 7(a) projects sample embeddings and prototypes via t-SNE, revealing that the two classes form distinguishable clusters with partial overlap. High-attention prototypes reside near cluster centers, serving as intra-cluster representatives. Low-attention prototypes scatter along inter-cluster boundaries, acting as boundary detectors. Figure 7(b) presents the average affinity weights of the top-10 most active prototypes on both classes. The prototypes split into two groups with no cross-activation. This pattern confirms that temperature-scaled Softmax and diversity regularization drive prototypes toward class-specific representations rather than collapse or sharing.

4.6 Runtime Analysis

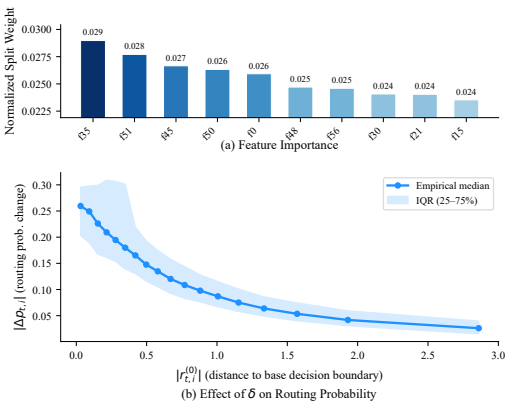


Figure 6: Visualization of NADT's internal decision mechanism on sonar. (a) Top-10 feature contributions measured by accumulated absolute split weights across all trees. (b) Impact of neighborhood bias δ on routing probability $|\Delta p_{t,i}|$ vs. distance from the decision boundary $|r_{t,i}^{(0)}|$.

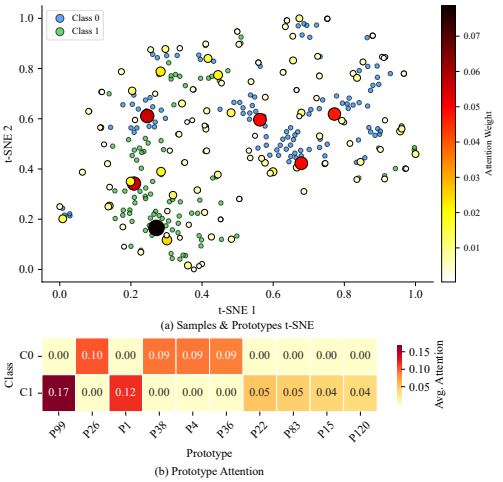


Figure 7: Visualization of prototype learning behavior on sonar. (a) t-SNE projection of sample embeddings and prototypes in the embedding space. (b) Top-10 most active prototypes with their average affinity weights on the two classes.

Figure 8 presents scatter plots of training versus inference time on two representative datasets(a HDLSS task and a moderate-scale multi-class task). Traditional tree ensembles train efficiently but inference speed varies by implementation. GRANDE, SoHoT, and ProtoGate incur notably higher costs in one or both dimensions, while ModernNCA forms an efficient tier. NADT consistently resides in the bottom-left region alongside ModernNCA. Across all 30 datasets, NADT completes inference below 0.1s on 29 of them, as NCM's prototype distance computation is independent of training-set size (incurring only $O(MD)$ work per sample) and involves no training-set retrieval. These results confirm that NADT maintains practical efficiency while providing neighborhood-aware routing.

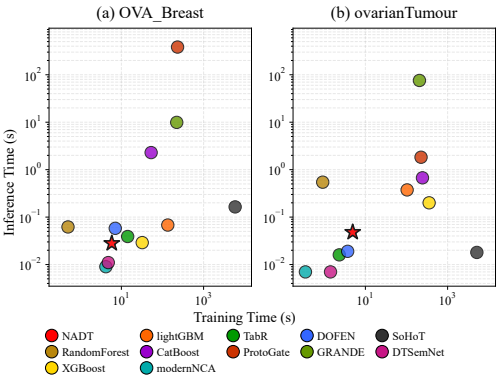


Figure 8: Training versus inference time on OVA_Breast and ovarianTumour.

5 Conclusion

In this paper, we proposed NADT, a tabular classification framework that integrates neighborhood awareness into the hierarchical routing of soft decision trees. The core innovation, the Neighborhood Context Module (NCM), uses learnable prototypes to generate sample-adaptive biases that shift split boundaries dynamically with each sample's neighborhood. These biases are injected into differentiable oblique splits of the soft decision tree ensemble, enabling neighborhood-aware routing without retrieving the training set at inference. Experiments demonstrate that NADT attains the best average rank among 11 competing methods and is the only neural tree within the statistically top-ranked tier. Ablation studies confirm that the performance gains stem from learning meaningful neighborhood structure. Visualization analysis reveals that prototypes learn class-specific representations and that the ensemble develops complementary routing patterns across trees. NADT also maintains efficiency in both training and inference. Future work includes exploring dynamic prototype allocation strategies, and integrating neighborhood context with other differentiable tree architectures.

AI USE STATEMENT

In this work, we used generative AI tools for English language polishing and spell checking, literature search and survey assistance, proofreading and checking of mathematical formulas, auxiliary implementation of utility functions in the model codebase, and visual style refinement of experimental result figures. We have not used generative AI tools for research idea generation, experimental design, or results analysis. All other tasks with required disclosure are not applicable to this work. We reviewed all AI-assisted outputs for correctness and appropriateness. We take full responsibility for the final content of this work, including all text, claims, and artifacts.

REPRODUCIBILITY STATEMENT

To facilitate reproducibility, we provide an anonymous source code repository containing the key implementation of NADT. The repository includes the hyperparameter values used for every method and dataset, as well as detailed default hyperparameter settings (Appendix E, Table 8). All 30 datasets used in our experiments are publicly available through OpenML (Vanschoren et al., 2014) and UCI repositories; their statistics and preprocessing steps are documented in Appendix D (Tables 2 and 3). Pseudocode of the NADT algorithm is provided in Appendix A, and implementation details including software versions and hardware configuration are described in Appendix G. Full experimental results across all datasets, including standard deviations and runtime measurements, are reported in Appendix F and Appendix I.

REFERENCES

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In Ankur Teredesai, Vipin Kumar, Ying Li, Rómer Rosales, Evimaria Terzi, and George Karypis (eds.), *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2623–2631. ACM, 2019.

Sercan Ö. Arik and Tomas Pfister. Tabnet: Attentive interpretable tabular learning. In *Thirty-Fifth AAAI Conference on Artificial Intelligence*, pp. 6679–6687. AAAI Press, 2021.

Yoshua Bengio, Nicolas Léonard, and Aaron C. Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *CoRR*, abs/1308.3432, 2013. URL http://arxiv.org/abs/1308.3432.

Vadim Borisov, Tobias Leemann, Kathrin Seßler, Johannes Haug, Martin Pawelczyk, and Gjergji Kasneci. Deep neural networks and tabular data: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 35(6):7499–7519, 2024.

Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

Kuan-Yu Chen, Ping-Han Chiang, Hsin-Rung Chou, Chih-Sheng Chen, and Tien-Hao Chang. DOFEN: deep oblivious forest ensemble. In *Advances in Neural Information Processing Systems 38*, 2024.

Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794. ACM, 2016.

Antonio Consolo, Edoardo Amaldi, and Emilio Carrizosa. Soft decision trees for survival analysis. *Eur. J. Oper. Res.*, 331(3):925–940, 2026.

Yury Gorishniy, Ivan Rubachev, and Artem Babenko. On embeddings for numerical features in tabular deep learning. In *Advances in Neural Information Processing Systems 35*, 2022.

Yury Gorishniy, Ivan Rubachev, Nikolay Kartashev, Daniil Shlenskii, Akim Kotelnikov, and Artem Babenko. Tabr: Tabular deep learning meets nearest neighbors. In *The Twelfth International Conference on Learning Representations*, 2024.

Yury Gorishniy, Akim Kotelnikov, and Artem Babenko. Tabm: Advancing tabular deep learning with parameter-efficient ensembling. In *The Thirteenth International Conference on Learning Representations*, 2025.

Léo Grinsztajn, Edouard Oyallon, and Gaël Varoquaux. Why do tree-based models still outperform deep learning on typical tabular data? In *Advances in Neural Information Processing Systems 35*, 2022.

Beimnet Bekele Guta. Localized TabICLv2: Scaling tabular in-context learning through k-nn. In *Proceedings of the 2nd ICML Workshop on Foundation Models for Structured Data (FMSD)*, Seoul, South Korea, 2026.

Laurent Hyafil and Ronald L. Rivest. Constructing optimal binary decision trees is np-complete. *Inf. Process. Lett.*, 5(1):15–17, 1976.

Xiangjian Jiang, Andrei Margeloiu, Nikola Simidjievski, and Mateja Jamnik. Protogate: Prototype-based neural networks with global-to-local feature selection for tabular biomedical data. In *Forty-first International Conference on Machine Learning*, 2024.

Jacek Karolczak and Jerzy Stefanowski. An interpretable prototype parts-based neural network for medical tabular data. In *Proceedings of the Second Workshop on Second Workshop on Explainable Artificial Intelligence for the Medical Domain (EXPLIMED 2025) co-located with 28th European Conference on Artificial Intelligence (ECAI 2025)*, volume 4059 of *CEUR Workshop Proceedings*, 2025.

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems 30*, pp. 3146–3154, 2017.

Kirsten Köbschall, Lisa Hartung, and Stefan Kramer. Adaptive differentiable trees for transparent learning on data streams. *Mach. Learn.*, 114(11):253, 2025.

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.

Guang-He Lee and Tommi S. Jaakkola. Oblique decision trees from derivatives of relu networks. In *8th International Conference on Learning Representations*, 2020.

Yunze Leng, Rohan Ghosh, and Mehul Motani. Tab-pet: Graph-based positional encodings for tabular transformers. In Sven Koenig, Chad Jenkins, and Matthew E. Taylor (eds.), *Fortieth AAAI Conference on Artificial Intelligence, Thirty-Eighth Conference on Innovative Applications of Artificial Intelligence, Sixteenth Symposium on Educational Advances in Artificial Intelligence*, pp. 22823–22832. AAAI Press, 2026.

Haoling Li, Jie Song, Mengqi Xue, Haofei Zhang, and Mingli Song. A survey of neural trees: Co-evolving neural networks and decision trees. *IEEE Transactions on Neural Networks and Learning Systems*, 36(7):11718–11737, 2025.

Hongyi Li, Han Lin, and Jun Xu. Hinge regression tree: A newton method for oblique regression tree splitting. In *International Conference on Learning Representations (ICLR)*, 2026.

Jiaqi Luo and Shixin Xu. Ncart: Neural classification and regression tree for tabular data. *Pattern Recognition*, 154:110578, 2024.

Qiangqiang Mao, Jiayang Ren, Yixiu Wang, Chenxuanyin Zou, Jingjing Zheng, and Yankai Cao. Differentiable decision tree via "relu+argmin" reformulation. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, pp. 166874–166909. Curran Associates, Inc., 2025.

Sascha Marton, Stefan Lüdtke, Christian Bartelt, and Heiner Stuckenschmidt. Gradtree: Learning axis-aligned decision trees with gradient descent. In *Thirty-Eighth AAAI Conference on Artificial Intelligence*, pp. 14323–14331. AAAI Press, 2024a.

Sascha Marton, Stefan Lüdtke, Christian Bartelt, and Heiner Stuckenschmidt. Grande: Gradient-based decision tree ensembles for tabular data. In *The Twelfth International Conference on Learning Representations*, 2024b.

Sreerama K. Murthy, Simon Kasif, and Steven Salzberg. A system for induction of oblique decision trees. *J. Artif. Intell. Res.*, 2:1–32, 1994.

Youssef Nader, Leon Sixt, and Tim Landgraf. DNNR: differential nearest neighbors regression. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato (eds.), *International Conference on Machine Learning, ICML*, volume 162 of *Proceedings of Machine Learning Research*, pp. 16296–16317. PMLR, 2022.

Meike Nauta, Ron van Bree, and Christin Seifert. Neural prototype trees for interpretable fine-grained image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pp. 14933–14943. Computer Vision Foundation / IEEE, 2021.

Subrat Prasad Panda, Blaise Genest, and Arvind Easwaran. Approximation-free differentiable oblique decision trees. *Journal of Machine Learning Research*, 27(125):1–31, 2026.

Deepak Pandita, Flip Korn, Chris Welty, and Christopher M. Homan. Forest vs tree: The (n, K) trade-off in reproducible ML evaluation. In Sven Koenig, Chad Jenkins, and Matthew E. Taylor (eds.), *Fortieth AAAI Conference on Artificial Intelligence*, pp. 24736–24744, 2026.

Sergei Popov, Stanislav Morozov, and Artem Babenko. Neural oblivious decision ensembles for deep learning on tabular data. In *8th International Conference on Learning Representations*, 2020.

Liudmila Ostroumova Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. Catboost: unbiased boosting with categorical features. In *Advances in Neural Information Processing Systems 31*, 2018.

Ravid Shwartz-Ziv and Amitai Armon. Tabular data: Deep learning is not all you need. *Information Fusion*, 81:84–90, 2022.

Jake Snell, Kevin Swersky, and Richard S. Zemel. Prototypical networks for few-shot learning. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems*, pp. 4077–4087, 2017.

Shriyank Somvanshi, Subasish Das, Syed Javed, Gian Antariksa, and Ahmed Hossain. A survey on tabular data: From tree-based methods to tabular deep learning. *ACM Computing Surveys*, April 2026.

Joaquin Vanschoren, Jan N Van Rijn, Bernd Bischl, and Luis Torgo. Openml: networked science in machine learning. *ACM SIGKDD Explorations Newsletter*, 15(2):49–60, 2014.

Yeming Wen, Dustin Tran, and Jimmy Ba. Batchensemble: an alternative approach to efficient ensemble and lifelong learning. In *8th International Conference on Learning Representations, ICLR*, 2020.

Han-Jia Ye, Huai-Hong Yin, De-Chuan Zhan, and Wei-Lun Chao. Revisiting nearest neighbor for tabular data: A deep tabular baseline two decades later. In *The Thirteenth International Conference on Learning Representations, ICLR*. OpenReview.net, 2025.

A Pseudo-Code of NADT

B Proofs for Theoretical Analysis

This section provides the complete proofs for the theoretical results stated in Section 3.5 of the main paper.

Algorithm 1 Training of Neighborhood-Augmented Neural Decision Trees (NADT)

Require: Training set $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$; hyperparameters T, depth, M, D, r, λ_{div}, m
Ensure: Trained NADT model

Initialize feature projector $\mathbf{W}_{\text{proj}}$, $\mathbf{b}_{\text{proj}}$
Initialize M prototypes $\{\mathbf{p}_m\}_{m=1}^{M}$ via Mini-Batch K-means on $\{\mathbf{W}_{\text{proj}}\mathbf{x}_i + \mathbf{b}_{\text{proj}}\}$
Initialize learnable temperature τ
Initialize context projection $\mathbf{W}_{\text{ctx}}$
for each tree $t \in \{1, \ldots, T\}$ **do**
    Sample feature indices $\text{feat_idx}_t \subset \{1, \ldots, d\}$ with ratio r
    Initialize split weights $\mathbf{w}_t^{\text{split}}$, biases $\mathbf{b}_t^{\text{split}}$, leaf vectors $\{\mathbf{v}_{t,\ell}\}_{\ell=1}^{L}$
end for
for epoch $= 1$ to E **do**
    for each mini-batch $\mathcal{B} \subset \mathcal{D}$ **do**
        Feature Projection: $\mathbf{x}_{\text{proj}} = \mathbf{W}_{\text{proj}}\mathbf{x} + \mathbf{b}_{\text{proj}}$
        NCM: compute distances $d_m = \|\mathbf{x}_{\text{proj}} - \mathbf{p}_m\|_2^2$ for $m = 1, \ldots, M$
        Compute affinity weights $\mathbf{w} = \text{Softmax}(-d_1/\tau, \ldots, -d_M/\tau)$
        Compute node biases $\boldsymbol{\delta} = \text{LayerNorm}(\mathbf{W}_{\text{ctx}}\mathbf{w}) \in \mathbb{R}^{T \times I}$
        Tree Ensemble:
        for each tree $t \in \{1, \ldots, T\}$ **do**
            $\mathbf{x}_{\text{sub}}^{(t)} = \mathbf{x}[\text{feat_idx}_t]$
            for each internal node i **do**
                $r_{t,i} = \mathbf{w}_{t,i}^{\text{split}\top}\mathbf{x}_{\text{sub}}^{(t)} + b_{t,i}^{\text{split}} + \delta_{t,i}$
                $p_{t,i} = (\text{softsign}(r_{t,i}) + 1)/2$
            end for
            Compute leaf reach probabilities $\mu_{t,\ell}$ via product of $p_{t,i}$ along path
            $\hat{\mathbf{y}}_t = \sum_{\ell=1}^{L} \mu_{t,\ell} \cdot \mathbf{v}_{t,\ell}$
        end for
        $\hat{\mathbf{y}} = \frac{1}{T}\sum_{t=1}^{T} \hat{\mathbf{y}}_t$
        $\mathcal{L}_{\text{CE}} = \text{CrossEntropy}(\hat{\mathbf{y}}, y)$
        $\mathcal{L}_{\text{div}} = \mathbb{E}_{a \neq b}\big[\max(0,\, m - \|\mathbf{p}_a - \mathbf{p}_b\|_2)\big]$
        $\mathcal{L} = \mathcal{L}_{\text{CE}} + \lambda_{\text{div}} \cdot \mathcal{L}_{\text{div}}$
        Backpropagate $\mathcal{L}$; clip gradients; update parameters via AdamW
    end for
end for

Algorithm 2 Inference of NADT

Require: Input sample $\mathbf{x}$; trained NADT model
Ensure: Predicted class $\hat{y}$

$\mathbf{x}_{\text{proj}} = \mathbf{W}_{\text{proj}}\mathbf{x} + \mathbf{b}_{\text{proj}}$
Compute $\boldsymbol{\delta} = \text{LayerNorm}(\mathbf{W}_{\text{ctx}} \cdot \text{Softmax}(-\|\mathbf{x}_{\text{proj}} - \mathbf{p}_m\|_2^2/\tau))$
for each tree $t \in \{1, \ldots, T\}$ **do**
    $\mathbf{x}_{\text{sub}}^{(t)} = \mathbf{x}[\text{feat_idx}_t]$
    for each internal node i **do**
        $r_{t,i} = \mathbf{w}_{t,i}^{\text{split}\top}\mathbf{x}_{\text{sub}}^{(t)} + b_{t,i}^{\text{split}} + \delta_{t,i}$
        $p_{t,i} = (\text{softsign}(r_{t,i}) + 1)/2$
    end for
    Compute leaf reach probabilities $\mu_{t,\ell}$
    $\hat{\mathbf{y}}_t = \sum_{\ell=1}^{L} \mu_{t,\ell} \cdot \mathbf{v}_{t,\ell}$
end for
$\hat{\mathbf{y}} = \frac{1}{T}\sum_{t=1}^{T} \hat{\mathbf{y}}_t$
$\hat{y} = \arg\max_c \hat{\mathbf{y}}_c$

B.1 Expressive Power of NCM Biases

Definition 1. *The* baseline function class *without NCM is*

$$\mathcal{F}_{base} = \left\{ f_0 : \mathbb{R}^d \to \mathbb{R}^C \;\middle|\; f_0(\mathbf{x}) = \frac{1}{T}\sum_{t=1}^{T}\sum_{\ell=1}^{L} \mu_{t,\ell}^{(0)}(\mathbf{x}) \cdot \mathbf{v}_{t,\ell} \right\}, \tag{11}$$

where $\mu_{t,\ell}^{(0)}$ *uses routing probabilities computed from* $r_{t,i}^{(0)} = \mathbf{w}_{t,i}^{split\top}\mathbf{x}_{sub}^{(t)} + b_{t,i}^{split}$.

Definition 2. *The* NADT function class *with NCM is*

$$\mathcal{F}_{NADT} = \left\{ f : \mathbb{R}^d \to \mathbb{R}^C \;\middle|\; f(\mathbf{x}) = \frac{1}{T}\sum_{t=1}^{T}\sum_{\ell=1}^{L} \mu_{t,\ell}(\mathbf{x}, \boldsymbol{\delta}(\mathbf{x})) \cdot \mathbf{v}_{t,\ell} \right\}, \tag{12}$$

where $\boldsymbol{\delta}(\mathbf{x}) = \mathrm{LN}(\mathbf{W}_{ctx}\mathbf{w}(\mathbf{x}))$ *and* $\mathbf{w}(\mathbf{x})$ *is the temperature-scaled Softmax affinity vector over* M *learnable prototypes.*

Lemma 4 (Non-triviality of the NCM mapping)**.** *When* $M \geq 2$ *and prototypes are distinct (guaranteed by the diversity loss* $\mathcal{L}_{div}$*), the affinity vector* $\mathbf{w}(\mathbf{x})$ *is non-constant over* $\mathbf{x}$*. Consequently, for any non-zero* $\mathbf{W}_{ctx}$*, there exist* $\mathbf{x}_1, \mathbf{x}_2$ *with* $\mathbf{W}_{ctx}\mathbf{w}(\mathbf{x}_1) \neq \mathbf{W}_{ctx}\mathbf{w}(\mathbf{x}_2)$.

Proof. Let $M = 2$ with $\mathbf{p}_1 \neq \mathbf{p}_2$. Setting $\mathbf{z}_1 = \mathbf{p}_1$ yields $w_1(\mathbf{p}_1) > 1/2$, while $\mathbf{z}_2 = \mathbf{p}_2$ yields $w_1(\mathbf{p}_2) < 1/2$. Thus $\mathbf{w}(\mathbf{p}_1) \neq \mathbf{w}(\mathbf{p}_2)$, and any non-zero $\mathbf{W}_{\text{ctx}}$ preserves this distinction. □

Theorem 5 (Strict function class containment)**.** $\mathcal{F}_{base} \subsetneq \mathcal{F}_{NADT}$.

Proof. Inclusion. Setting $\mathbf{W}_{\text{ctx}} = \mathbf{0}$ yields $\boldsymbol{\delta}(\mathbf{x}) \equiv \mathbf{0}$, recovering every function in $\mathcal{F}_{\text{base}}$.

Strictness. By Lemma 4, there exist parameters producing non-constant $\delta_{t,i}(\mathbf{x})$. Choose $\mathbf{x}_1, \mathbf{x}_2$ with $\delta_{t,i}(\mathbf{x}_1) \neq \delta_{t,i}(\mathbf{x}_2)$ and $r_{t,i}^{(0)}(\mathbf{x}_1) = r_{t,i}^{(0)}(\mathbf{x}_2) = a$. Since softsign is strictly monotone, $p_{t,i}(\mathbf{x}_1) = \frac{\mathrm{softsign}(a+\delta_{t,i}(\mathbf{x}_1))+1}{2} \neq \frac{\mathrm{softsign}(a+\delta_{t,i}(\mathbf{x}_2))+1}{2} = p_{t,i}(\mathbf{x}_2)$, yielding distinct leaf reach probabilities and classifiers unreachable by $\mathcal{F}_{\text{base}}$. □

Remark 1 (Local heterogeneity)**.** *When the data distribution exhibits local heterogeneity—different regions* $\mathcal{R}_1, \ldots, \mathcal{R}_K$ *requiring different optimal split offsets—NADT can apply region-specific biases* $\delta_{t,i}(\mathbf{x}) \approx \Delta_k$ *for* $\mathbf{x} \in \mathcal{R}_k$*, while the baseline applies a single global offset. This provides the theoretical basis for the boundary adaptation observed in Figure 1 of the main paper.*

Lemma 6 (Temperature-controlled localization)**.** *The localization of NCM biases is governed by the temperature* τ*. As* $\tau \to 0^+$*,* $w_m(\mathbf{x})$ *converges to hard nearest-neighbor assignment, making* $\boldsymbol{\delta}(\mathbf{x})$ *piecewise-constant over Voronoi cells of the prototypes. As* $\tau \to \infty$*,* $w_m(\mathbf{x}) \to 1/M$ *and* $\boldsymbol{\delta}$ *degenerates to a constant.*

Proof. Let $m^* = \arg\min_{m'} \|\mathbf{x} - \mathbf{p}_{m'}\|_2^2$. For $m \neq m^*$, the ratio $w_m/w_{m^*} = \exp\big(-(\|\mathbf{x}-\mathbf{p}_m\|_2^2 - \|\mathbf{x}-\mathbf{p}_{m^*}\|_2^2)/\tau\big) \to 0$ as $\tau \to 0^+$, so $w_{m^*} \to 1$. As $\tau \to \infty$, all exponents tend to zero, yielding the uniform distribution. □

B.2 Gradient Stability of Softsign Routing

Lemma 7 (Softsign gradient bounds)**.** *The softsign routing function* $\sigma_s(r) = r/(1+|r|)$ *satisfies* $\sigma_s'(r) = 1/(1+|r|)^2$*, so that* $\frac{1}{2(1+R)^2} \leq \frac{\partial p_{t,i}}{\partial r_{t,i}} \leq \frac{1}{2}$ *for all* $|r_{t,i}| \leq R$.

Proof. For $r \geq 0$: $\sigma_s(r) = r/(1+r)$, $\sigma_s'(r) = 1/(1+r)^2$. For $r < 0$: $\sigma_s(r) = r/(1-r)$, $\sigma_s'(r) = 1/(1-r)^2 = 1/(1+|r|)^2$. Since $p_{t,i} = (\sigma_s(r_{t,i})+1)/2$, the result follows by multiplying by $1/2$ and bounding $(1+|r|)^2 \in [1, (1+R)^2]$. □

Lemma 8 (Sigmoid gradient decay)**.** *The sigmoid routing probability* $p_{t,i}^{\text{sig}} = \sigma(r_{t,i})$ *satisfies* $\frac{\partial p_{t,i}^{\text{sig}}}{\partial r_{t,i}} = \sigma(r_{t,i})(1-\sigma(r_{t,i})) \leq e^{-|r_{t,i}|}$.

Proof. $\sigma(r)(1-\sigma(r)) = e^{-r}/(1+e^{-r})^2 \le e^{-|r|}$. □

Theorem 9 (Depth-independent log-path gradient)**.** *Let D be the tree depth and $|r_{t,i}| \le R$ for all internal nodes on the path to leaf ℓ. Then $\left|\partial \log \mu_{t,\ell}/\partial r_{t,i_k}\right| \ge 1/(2(1+R)^2)$ for any node i_k on the path, regardless of its depth k.*

Proof. Since $\log \mu_{t,\ell} = \sum_{j=1}^{D} \log \tilde{p}_{t,i_j}$, the partial derivative with respect to r_{t,i_k} isolates the k-th term. We analyze the two branch cases.

Right branch ($\tilde{p} = p$). For $r \ge 0$: $\frac{\partial \log p}{\partial r} = \frac{\sigma_s'(r)}{\sigma_s(r)+1} = \frac{1/(1+r)^2}{(1+2r)/(1+r)} = \frac{1}{(1+r)(1+2r)} \ge \frac{1}{2(1+r)^2}$. For $r < 0$: $\frac{\partial \log p}{\partial r} = \frac{1/(1-r)^2}{1/(1-r)} = \frac{1}{1-r} = \frac{1}{1+|r|} \ge \frac{1}{1+R}$.

Left branch ($\tilde{p} = 1-p$). By symmetry, $|\partial \log(1-p)/\partial r| = |\partial \log p/\partial r|\big|_{r\to -r}|$, yielding the same bound.

The overall minimum across all cases is $1/(2(1+R)^2)$, attained by the right-branch case at $r \ge 0$, where $\frac{1}{(1+r)(1+2r)} \ge \frac{1}{2(1+r)^2} \ge \frac{1}{2(1+R)^2}$. The remaining cases yield larger values: the right branch at $r < 0$ gives $1/(1+|r|) \ge 1/(1+R) \ge 1/(2(1+R)^2)$. This lower bound is therefore independent of D and k. □

Remark 2 (Comparison with sigmoid routing)**.** *For a like-for-like comparison, consider the log-path gradient. Under sigmoid routing with $p_{t,i}^{\text{sig}} = \sigma(r_{t,i})$, the right-branch term satisfies $\partial \log \mu_{t,\ell}^{\text{sig}}/\partial r_{t,i} = 1-\sigma(r_{t,i}) = \sigma(-r_{t,i})$, and the left-branch term has magnitude $\sigma(r_{t,i})$; both are bounded below by $\sigma(-R)$ whenever $|r_{t,i}| \le R$, so the sigmoid log-path gradient is likewise depth-independent. The two activations differ in how this lower bound decays with the margin R: $\sigma(-R) \le e^{-R}$ decays* exponentially*, whereas the softsign bound $1/(2(1+R)^2)$ in Theorem 9 decays only* polynomially*. Softsign therefore retains informative gradients for confident samples far from decision boundaries, supporting effective end-to-end training without special techniques such as the Straight-Through estimator (Bengio et al., 2013) or gradient surrogates used in sigmoid-routed trees (Kontschieder et al., 2015).*

Lemma 10 (Adaptive gradient focus of NCM biases)**.** *Since $r_{t,i} = r_{t,i}^{(0)} + \delta_{t,i}$, we have $\partial p_{t,i}/\partial \delta_{t,i} = 1/(2(1+|r_{t,i}|)^2)$. The gradient signal for NCM biases is maximized near decision boundaries ($|r_{t,i}| \approx 0$) and diminishes for confident samples ($|r_{t,i}| \gg 1$).*

Proof. Direct from the chain rule and Lemma 7. □

Remark 3 (Connection to visualization)**.** *Lemma 10 provides the theoretical explanation for the experimental observation in Figure 5(b) of the main paper: $|\Delta p_{t,i}|$ decreases monotonically with $|r_{t,i}^{(0)}|$, confirming that NCM biases naturally prioritize boundary-uncertain routing.*

C Synthetic Dataset for Decision Boundary Visualization

Real-world datasets often exhibit local heterogeneity, as different regions require different optimal decision boundary directions. Methods relying on globally fixed splits, global retrieval, or global metric learning cannot simultaneously align with all local geometries. We construct a binary classification dataset with three local regions. The three region centers are placed on a circle of radius $R = 1.4$, evenly spaced at 120° intervals forming a trident layout. Each region contains samples from both Class 0 and Class 1, with the optimal linear separation direction differing by 120° across regions (0°, 120°, and 240° respectively). Random noise ($\sigma = 0.16$) is added within each region so that local boundaries are not perfectly separable. Each region contributes 200 samples per class, yielding 1,200 samples in total, all of which are standardized using a StandardScaler. Background colors denote predicted class regions (orange-red for Class 0, blue for Class 1), black curves mark decision boundaries, and scatter points show training (semi-transparent small dots) and test (larger dots with white edges) samples.

D Dataset statistics and Experimental Setup

All experiments use public datasets from OpenML (Vanschoren et al., 2014)[3], with detailed statistics provided in Table 2. We select 30 datasets covering both binary and multi-class tasks. They exhibit wide variation across key statistics: sample sizes range from 60 to 99,289, feature dimensions from 4 to 54,621, class counts from 2 to 11, and imbalance ratios from 1.0 to 27.38. All input features are preprocessed with Min-Max normalization, scaling each feature to the $[0, 1]$ range.

Table 2: Dataset statistics.

Dataset	Samples	Features	Classes	Class Distribution (Counts and Proportions)	Imb. Ratio
tumors_C	60	7129	2	0:39 (65.0%), 1:21 (35.0%)	1.86
AP_Endometrium_Prostate	130	10935	2	0:61 (46.9%), 1:69 (53.1%)	1.13
AP_Omentum_Lung	203	10935	2	0:126 (62.1%), 1:77 (37.9%)	1.64
sonar	208	60	2	0:111 (53.4%), 1:97 (46.6%)	1.14
thyroid	215	5	3	0:150 (69.8%), 1:35 (16.3%), 2:30 (14.0%)	5.00
BurkittLymphoma	220	22283	3	0:48 (21.8%), 1:44 (20.0%), 2:128 (58.2%)	2.91
AP_Endometrium_Ovary	259	10935	2	0:61 (23.6%), 1:198 (76.4%)	3.25
ovarianTumour	283	54621	3	0:18 (6.4%), 1:20 (7.1%), 2:245 (86.6%)	13.61
AP_Omentum_Kidney	337	10935	2	0:260 (77.2%), 1:77 (22.9%)	3.38
AP_Uterus_Kidney	384	10935	2	0:260 (67.7%), 1:124 (32.3%)	2.10
AP_Breast_Omentum	421	10935	2	0:344 (81.7%), 1:77 (18.3%)	4.47
AP_Colon_Kidney	546	10935	2	0:286 (52.4%), 1:260 (47.6%)	1.10
eating	945	6373	7	0:140 (14.8%), 1:140 (14.8%), 2:133 (14.1%), 3:140 (14.8%), 4:119 (12.6%), 5:133 (14.1%), 6:140 (14.8%)	1.18
cortex_nuclear	1077	68	8	0:150 (13.9%), 1:135 (12.5%), 2:150 (13.9%), 3:135 (12.5%), 4:135 (12.5%), 5:105 (9.8%), 6:135 (12.5%), 7:132 (12.3%)	1.43
banknote-authentication	1372	4	2	0:762 (55.5%), 1:610 (44.5%)	1.25
OVA_Breast	1545	10935	2	0:344 (22.3%), 1:1201 (77.7%)	3.49
OVA_Uterus	1545	10935	2	0:1421 (92.0%), 1:124 (8.0%)	11.46
OVA_Kidney	1545	10935	2	0:260 (16.8%), 1:1285 (83.2%)	4.94
OVA_Lung	1545	10935	2	0:1421 (92.0%), 1:124 (8.0%)	11.46
OVA_Omentum	1545	10935	2	0:77 (5.0%), 1:1468 (95.0%)	19.06
OVA_Endometrium	1545	10935	2	0:61 (4.0%), 1:1484 (96.0%)	24.33
OVA_Prostate	1545	10935	2	0:1476 (95.5%), 1:69 (4.5%)	21.39
pollen	3848	5	2	0:1924 (50.0%), 1:1924 (50.0%)	1.00
hiva_agnostic	4229	1617	2	0:4080 (96.5%), 1:149 (3.5%)	27.38
texture	5500	40	11	0:500 (9.1%), 1:500 (9.1%), 2:500 (9.1%), 3:500 (9.1%), 4:500 (9.1%), 5:500 (9.1%), 6:500 (9.1%), 7:500 (9.1%), 8:500 (9.1%), 9:500 (9.1%), 10:500 (9.1%)	1.00
optdigits	5620	64	10	0:554 (9.9%), 1:571 (10.2%), 2:557 (9.9%), 3:572 (10.2%), 4:568 (10.1%), 5:558 (9.9%), 6:558 (9.9%), 7:566 (10.1%), 8:554 (9.9%), 9:562 (10.0%)	1.03
JapaneseVowels	9961	14	2	0:8347 (83.8%), 1:1614 (16.2%)	5.17
elevators	16599	18	2	0:5130 (30.9%), 1:11469 (69.1%)	2.24
jannis	83733	54	4	0:1687 (2.0%), 1:28790 (34.4%), 2:14734 (17.6%), 3:38522 (46.0%)	22.83
SVHN	99289	3072	10	0:18960 (19.1%), 1:6692 (6.7%), 2:14734 (14.8%), 3:11379 (11.5%), 4:9981 (10.1%), 5:9266 (9.3%), 6:7704 (7.8%), 7:7614 (7.7%), 8:6705 (6.8%), 9:6254 (6.3%)	3.03

We use F1(Macro) as the primary evaluation metric to comprehensively assess model performance across all classes. The F1 score is defined as the harmonic mean of precision and recall:

$$\mathrm{F1} = \frac{2 \times \mathrm{TP}}{2 \times \mathrm{TP} + \mathrm{FP} + \mathrm{FN}} \tag{13}$$

[3] https://www.openml.org/

where TP, FP, and FN denote true positives, false positives, and false negatives, respectively. Macro averaging computes F1 for each class independently and then takes the arithmetic mean, giving equal weight to every class and preventing majority classes from dominating the evaluation.

NADT is implemented in Python using the PyTorch framework. All experiments are conducted on a Windows 11 server with the following hardware configuration: Intel i9-13900K CPU, 64GB RAM, and an NVIDIA GeForce RTX 4090 GPU.

We fix the base random seed to 42 and increment it by the repeat index across the five independent experimental runs, while applying the same seed simultaneously to NumPy, PyTorch CPU, and PyTorch CUDA backends with `torch.backends.cudnn. deterministic = True`. The data splitting is performed via `RepeatedStratifiedKFold` with a fixed `random_state`. Table 3 reports the standard deviation of F1(Macro) scores across 25 repeated stratified 5-fold runs.

Table 3: Standard deviation of F1(Macro) scores across 30 datasets (5×5 cross-validation). "—" indicates that the method could not run on the dataset.

Dataset	NADT	RF	XGBoost	LGBM	CatBoost	MNCA	TabR	P.Gate	DOFEN	GRANDE	SoHoT	DTSemNet
thyroid	0.0391	0.0472	0.0416	0.0399	0.0407	0.0385	0.0373	0.0632	0.0000	0.0429	0.0932	0.0553
cortex_nuclear	0.0042	0.0046	0.0117	0.0083	0.0078	0.0043	0.0075	0.0058	0.0413	0.0198	0.0665	0.0092
optdigits	0.0037	0.0043	0.0038	0.0044	0.0043	0.0026	0.0038	—	0.0032	0.0041	0.0734	0.0085
texture	0.0018	0.0038	0.0029	0.0035	0.0043	0.0006	0.0009	—	0.0040	0.0072	0.0706	0.0086
banknote-authentication	0.0000	0.0051	0.0033	0.0052	0.0040	0.0024	0.0024	0.0015	0.0037	0.0071	0.0161	0.0073
sonar	0.0531	0.0734	0.0752	0.0726	0.0697	0.0513	0.0429	0.0416	0.0948	0.0732	0.0611	0.0482
pollen	0.0152	0.0162	0.0163	0.0165	0.0145	0.0137	0.0379	—	0.0008	0.0236	0.0190	0.0160
elevators	0.0062	0.0091	0.0086	0.0086	0.0083	0.0041	0.0054	—	0.0113	0.0087	0.0067	0.0059
JapaneseVowels	0.0025	0.0042	0.0033	0.0029	0.0036	0.0019	0.0013	—	0.0070	0.0057	0.0127	0.0043
AP_Colon_Kidney	0.0131	0.0145	0.0138	0.0142	0.0130	0.0141	0.0138	0.0157	0.0072	0.0130	0.0050	0.0156
AP_Endo._Prostate	0.0078	0.0193	0.0236	0.0401	0.0224	0.0145	0.0177	0.0240	0.0526	0.0200	0.0707	0.0285
AP_Breast_Omentum	0.0289	0.0315	0.0298	0.0276	0.0308	0.0307	0.0308	0.0372	0.0017	0.0429	0.0480	0.0458
AP_Omentum_Kidney	0.0276	0.0361	0.0362	0.0365	0.0348	0.0220	0.0284	0.0236	0.0019	0.0272	0.0323	0.0368
AP_Omentum_Lung	0.0368	0.0433	0.0532	0.0424	0.0447	0.0342	0.0389	0.0574	0.0038	0.0421	0.0134	0.0722
AP_Uterus_Kidney	0.0193	0.0247	0.0245	0.0234	0.0242	0.0169	0.0240	0.0214	0.0013	0.0217	0.0352	0.0300
AP_Endo._Ovary	0.0529	0.0649	0.0592	0.0644	0.0577	0.0479	0.0369	0.0561	0.0025	0.0395	0.0674	0.0608
hiva_agnostic	0.0435	0.0498	0.0526	0.0511	0.0495	0.0392	0.0395	—	0.0001	0.0532	0.0001	0.0356
eating	0.0258	0.0324	0.0359	0.0356	0.0487	0.0330	0.0281	0.0311	0.0620	0.0228	0.0165	0.0301
tumors_C	0.1663	0.1276	0.1429	0.1505	0.1313	0.1593	0.1349	0.1658	0.0133	0.1213	0.0141	0.1516
ovarianTumour	0.0998	0.0674	0.0888	0.0869	0.0620	0.0991	0.0742	0.0575	0.0015	0.1321	0.0016	0.0964
BurkittLymphoma	0.0443	0.0804	0.0592	0.0552	0.0774	0.0628	0.0646	0.0598	0.0031	0.0630	0.0033	0.0748
OVA_Breast	0.0139	0.0160	0.0115	0.0121	0.0132	0.0155	0.0137	0.0162	0.0004	0.0137	0.0248	0.0181
OVA_Uterus	0.0443	0.0477	0.0374	0.0423	0.0532	0.0436	0.0512	0.0412	0.0004	0.0411	0.0932	0.0393
OVA_Kidney	0.0089	0.0119	0.0118	0.0112	0.0086	0.0096	0.0118	0.0118	0.0000	0.0118	0.0238	0.0126
OVA_Lung	0.0443	0.0482	0.0425	0.0433	0.0496	0.0461	0.0372	0.0441	0.0004	0.0411	0.0932	0.0393
OVA_Omentum	0.0636	0.0282	0.0770	0.0632	0.0613	0.0512	0.0648	0.0708	0.0004	0.0734	0.0004	0.0645
OVA_Endometrium	0.0595	0.0156	0.0745	0.0792	0.0840	0.0632	0.0710	0.0820	0.0004	0.0633	0.0004	0.0602
OVA_Prostate	0.0096	0.0137	0.0174	0.0171	0.0135	0.0187	0.0114	0.0146	0.0004	0.0335	0.0255	0.0266
jannis	0.0062	0.0026	0.0060	0.0068	0.0049	0.0044	0.0092	—	0.0073	0.0065	0.0027	0.0058
SVHN	0.0019	0.0030	0.0039	0.0034	0.0054	0.0027	0.0027	—	0.0029	0.0047	0.0158	0.0265

E Baseline Methods and Hyperparameter Search Spaces

We compare NADT against 11 baseline methods spanning traditional tree ensembles and deep learning approaches. Brief descriptions of each method are provided below.

Random Forest (RF)[4] (Breiman, 2001) constructs an ensemble of decision trees via bagging. Each tree is trained on an independent bootstrap sample with random feature subsampling, and predictions are aggregated by majority voting. RF is robust to overfitting due to the randomness injected during training.

XGBoost[5] (Chen & Guestrin, 2016) implements gradient-boosted decision trees with second-order Taylor approximation of the loss. It introduces regularized learning objectives, sparsity-aware split finding, and column-block parallelism. Trees are added sequentially to fit the negative gradient of the loss.

[4] `https://scikit-learn.org/stable/modules/ensemble.html#forest`
[5] `https://github.com/dmlc/xgboost`

LightGBM[6] (Ke et al., 2017) accelerates gradient boosting through Gradient-based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB).

CatBoost[7] (Prokhorenkova et al., 2018) addresses prediction shift through Ordered Boosting and handles categorical features natively via ordered target statistics. It uses oblivious symmetric trees as base learners, where all nodes at the same depth share the same split condition.

TabR[8] (Gorishniy et al., 2024) is a retrieval-augmented feed-forward network that inserts a single-head retrieval module after an MLP encoder, using Key-only L2 distance and label-aware Value correction for prediction.

GRANDE[9] (Marton et al., 2024b) trains an end-to-end gradient-based ensemble of hard axis-aligned decision trees, using a dense tree representation with Straight-Through estimator for differentiable hard splits.

DOFEN[10] (Chen et al., 2024) is a deep oblivious forest ensemble that achieves sparse column selection via random enumeration of ODT feature combinations with differentiable weighting.

ModernNCA[11] (Ye et al., 2025) modernizes Neighborhood Components Analysis with deep MLP embeddings, soft-NN loss, and Stochastic Neighborhood Sampling.

ProtoGate[12] (Jiang et al., 2024) is a prototype-based neural network combining global-to-local feature selection with a non-parametric prototype KNN predictor.

SoHoT[13] (Köbschall et al., 2025) is a differentiable decision tree for data streams that combines Hoeffding Tree incremental growth with soft routing.

DTSemNet[14] (Panda et al., 2026) encodes hard oblique decision trees as semantically equivalent shallow ReLU networks, enabling vanilla gradient descent without Straight-Through estimators. Internal nodes use linear+ReLU pairs, while leaf nodes use fixed 0/1 connections for path constraints.

Hyperparameters of all baseline methods are tuned via Optuna on a holdout validation set (3:1:1 split), with search spaces listed in Table 4.

F Ablation Variants and Results

F.1 Ablation Setup

To validate the contribution of each component in NADT, we design four ablation variants and one capacity-matched control variant. All variants share the same default hyperparameters ($T = 128$, depth$= 4$, $M = 256$, $\lambda_{\text{div}} = 0.001$, $r = 0.5$) and training protocol.

NADT (Full). The complete model serves as the baseline, consisting of a feature projector, a Neighborhood Context Module (NCM), and a soft decision tree ensemble. The feature projector maps raw inputs to a D-dimensional embedding space. NCM computes squared Euclidean distances between the projected embedding and M learnable prototypes, then converts these distances to affinity weights via temperature-scaled Softmax. The resulting weights are mapped through a linear projection with LayerNorm to produce node biases $\boldsymbol{\delta} \in \mathbb{R}^{T \times I}$ for every internal node of every tree. The soft decision tree performs oblique splits on F_{sub}-dimensional feature subsets, with NCM biases injected into the split activation. Prototypes are initialized via K-means, and the model is trained end-to-end with cross-entropy and prototype diversity loss.

[6] https://github.com/microsoft/LightGBM
[7] https://github.com/catboost/catboost
[8] https://github.com/yandex-research/tabr
[9] https://github.com/s-marton/GRANDE
[10] https://github.com/Sinopac-Digital-Technology-Division/DOFEN
[11] https://github.com/LAMDA-Tabular/TALENT
[12] https://github.com/SilenceX12138/ProtoGate
[13] https://github.com/kramerlab/SoHoT
[14] https://github.com/CPS-research-group/dtsennet

Table 4: Hyperparameter search spaces of all baseline methods. All methods use Optuna with TPE sampler.

Method	Hyperparameter	Search Space
Random Forest	n_estimators	{50, 100, 150}
	max_depth	{None, 5, 10}
	min_samples_leaf	{1, 3}
XGBoost	n_estimators	{50, 100, 150}
	max_depth	$[3, 6]$ (int)
	learning_rate	{0.05, 0.1, 0.2}
LightGBM	n_estimators	{50, 100, 150}
	num_leaves	{15, 31, 63}
	learning_rate	{0.05, 0.1, 0.2}
CatBoost	iterations	{50, 100, 150}
	depth	$[4, 8]$ (int)
	learning_rate	{0.05, 0.1, 0.2}
TabR	d_main	{64, 128, 256}
	d_multiplier	{1.0, 2.0}
	encoder_n_blocks	$[0, 2]$ (int)
	predictor_n_blocks	$[1, 2]$ (int)
	context_dropout	$[0.0, 0.3]$ (float)
	dropout0	$[0.0, 0.3]$ (float)
	lr	$[10^{-4}, 5\times10^{-3}]$ (log)
	weight_decay	$[10^{-6}, 10^{-3}]$ (log)
	context_size	{32, 64, 96}
GRANDE	depth	$[3, 6]$ (int)
	n_estimators	{50, 100, 150}
	dropout	{0.0, 0.25}
DOFEN	m (N_cond = m×d)	{16, 32}
	d (rODT depth)	{4, 6}
	n_forest	{50, 100, 150}
	n_hidden	{64, 128, 256}
	dropout	{0.0, 0.1}
ModernNCA	dim	{128, 256, 512}
	d_block	{32, 64, 128}
	n_blocks	$[0, 2]$ (int)
	dropout	$[0.0, 0.5]$ step 0.1
	lr	$\{10^{-4}, 3\times10^{-4}, 10^{-3}\}$
	weight_decay	$\{0.0, 10^{-5}, 10^{-4}\}$
	temperature	{0.5, 1.0, 2.0}
	sample_rate	{0.5, 0.8, 1.0}
ProtoGate	lr	$\{10^{-4}, 3\times10^{-4}, 10^{-3}\}$
	lam_global	$\{10^{-4}, 2\times10^{-4}, 5\times10^{-4}\}$
	lam_local	$\{5\times10^{-4}, 10^{-3}, 2\times10^{-3}\}$
	pred_k	{3, 5}
	gating_hidden	{[100], [200]}
	sigma	{0.3, 0.5}
SoHoT	max_depth	$[3, 8]$ (int)
	lr	$[10^{-4}, 10^{-2}]$ (log)
	batch_size	{32, 64, 128}
	epochs	$[20, 60]$ step 10
	alpha	$[0.1, 0.9]$ step 0.2
	grace_period	{200, 400, 600}
DTSemNet	height	$[3, 6]$ (int)
	lr	$[10^{-4}, 10^{-2}]$ (log)
	batch_size	{32, 64, 128}
	epochs	$[30, 100]$ step 10

NADT w/o NCM. The NCM module is removed, setting $\boldsymbol{\delta} = \mathbf{0}$. This variant reduces to a plain soft decision tree ensemble: the split activation is determined solely by $r_{t,i} = \mathbf{w}_{t,i}^{\text{split}} \cdot \mathbf{x}_{\text{sub}}^{(t)} + b_{t,i}^{\text{split}}$, with no sample-adaptive neighborhood correction. The feature projector and prototype diversity loss are also removed, as they are only meaningful when NCM is present. If removing NCM causes a substantial performance drop, it confirms that sample-adaptive routing biases are critical to NADT.

NADT w/ Random $\boldsymbol{\delta}$. NCM is replaced with a fixed random projection: the feature projector is retained but frozen, and a frozen random linear layer $\mathbb{R}^D \to \mathbb{R}^{T \times I}$ directly generates $\boldsymbol{\delta}$ from the embedding. All NCM-related parameters are frozen after initialization. This variant retains the same parameter scale and bias injection point as the full model, but the bias signals are random and non-learnable. Comparing this variant with the full model distinguishes whether NCM gains come from learned neighborhood structure or merely from additional random perturbation acting as regularization.

NADT w/o Projector. The feature projector is removed, and NCM computes prototype distances directly in the raw input space. Prototype dimensionality changes from D to d_{in}, and K-means initialization is performed in the raw feature space. The soft decision trees still use raw feature subsets for splits. If removing the projector causes performance degradation, it indicates that compressing high-dimensional raw features into a low-dimensional embedding is crucial for meaningful distance metrics in NCM.

NADT w/ Axis-Aligned Splits. Oblique splits are replaced with axis-aligned threshold splits. Instead of using a weighted combination of F_{sub} features, each internal node only thresholds a single input feature. The split feature index for each (t, i) pair is randomly fixed at initialization, and the node learns only a threshold parameter $b_{t,i}$, producing $r_{t,i} = x[\text{feat_idx}(t, i)] + b_{t,i} + \delta_{t,i}$. NCM and the feature projector remain unchanged.

NADT w/ MLP Router (capacity-matched control). NCM is replaced by a plain two-layer MLP while the feature projector is retained. The MLP maps the projected embedding directly to the node biases via $\mathbb{R}^D \to \text{Linear}(D, H) \to \text{ReLU} \to \text{Linear}(H, T \times I)$. The hidden width H is solved so that the MLP has exactly the same number of trainable parameters as the NCM module it replaces. Since this control preserves the input, the bias injection point, and the parameter budget. It isolates whether NCM's gains stem from the prototype-based neighborhood structure itself or merely from any learnable input-conditioned routing. The prototype diversity loss is removed, as no prototypes are present.

Table 5: Component comparison of NADT and its ablation variants.

Component	Variant					
	Full	w/o NCM	w/ Rand $\boldsymbol{\delta}$	w/o Proj.	Axis-Aligned	MLP Router
Feature Projector	✓	–	✓*	–	✓	✓
NCM (prototype-based $\boldsymbol{\delta}$)	✓	–	–	✓	✓	–
Oblique Splits	✓	✓	✓	✓	–	✓
Diversity Loss $\mathcal{L}_{\text{div}}$	✓	–	–	✓	✓	–

*Frozen after initialization (non-trainable).

F.2 Ablation Results

Table 6 reports the F1(Macro) scores of all six variants. NADT (Full) achieves the best average rank (2.37), outperforming the MLP Router on 21 of 30 datasets, demonstrating the collective benefit of all components. Removing NCM causes severe degradation on several datasets, with F1 dropping from 0.9436 to 0.2740 on thyroid, confirming that sample-adaptive biases are critical for complex decision boundaries. The Random $\boldsymbol{\delta}$ variant performs similarly to w/o NCM, indicating that NCM's gains come from learned neighborhood structure rather than random perturbation. The MLP Router control matches NCM's parameter count but routes through a plain MLP, attaining an average rank of 3.60, identical to the Random $\boldsymbol{\delta}$ variant. It collapses precisely where prototypes matter most, dropping from 0.9969 to 0.7281 on cortex_nuclear and from 0.6852 to 0.5762 on hiva_agnostic. The markedly higher run-to-run variance (std 0.1028 vs. 0.0047 on cortex_nuclear) confirms that the gains stem from the prototype-based neighborhood structure rather than generic input-conditioned

routing. Removing the projector has the mildest impact overall, with noticeable drops on high-dimensional datasets such as SVHN (0.8277 $\rightarrow$ 0.7739), suggesting that the projector alleviates the curse of dimensionality for distance metrics. The Axis-Aligned variant suffers the most severe degradation across nearly all datasets. Performance on ovarianTumour drops from 0.6796 to 0.3183, and on OVA_Endometrium from 0.7070 to 0.5470, confirming that oblique splits are indispensable for capturing complex feature interactions.

Table 6: F1(Macro) scores of NADT and its ablation variants across 30 datasets. Best results are in bold; second-best are underlined.

Dataset	NADT	w/o NCM	w/ Rand $\boldsymbol{\delta}$	w/o Proj.	Axis-Aligned	MLP Router
thyroid	0.9436	0.2740	0.2740	0.9287	0.9407	**0.9480**
cortex_nuclear	0.9969	0.6579	0.7053	0.9941	**0.9972**	0.7281
optdigits	0.9804	0.9581	0.9576	0.9777	**0.9807**	0.9711
texture	**0.9979**	0.9224	0.9289	0.9920	0.9977	0.9722
banknote-authentication	**1.0000**	0.9738	0.9713	0.9994	**1.0000**	**1.0000**
sonar	0.8655	0.7853	0.7987	0.8426	**0.8753**	0.8292
pollen	0.4935	0.4841	0.4821	**0.4992**	0.4868	0.4880
elevators	0.8531	0.8704	0.8712	0.8371	0.8522	**0.8869**
JapaneseVowels	0.9926	0.9637	0.9644	**0.9933**	0.9931	0.9769
AP_Colon_Kidney	0.9791	0.9797	**0.9805**	0.9804	0.9699	0.9789
AP_Endometrium_Prostate	0.9953	**0.9969**	0.9966	0.9961	0.9799	0.9884
AP_Breast_Omentum	**0.9405**	0.9366	0.9363	0.9388	0.9129	0.9205
AP_Omentum_Kidney	0.9635	**0.9651**	0.9628	0.9629	0.9465	0.9641
AP_Omentum_Lung	0.9155	0.9152	0.9141	0.9126	0.8749	**0.9217**
AP_Uterus_Kidney	0.9725	0.9718	0.9712	0.9718	0.9584	**0.9778**
AP_Endometrium_Ovary	0.8900	0.8887	**0.8904**	0.8892	0.8513	0.8785
hiva_agnostic	**0.6852**	0.6613	0.6576	0.6828	0.6495	0.5762
eating	0.6371	0.6334	0.6318	0.6364	0.5807	**0.6427**
tumors_C	0.5649	**0.5974**	0.5778	0.5566	0.5370	0.5329
ovarianTumour	0.6796	0.6732	0.6724	**0.6831**	0.3183	0.6281
BurkittLymphoma	**0.8545**	0.8488	0.8495	0.8420	0.7409	0.8523
OVA_Breast	**0.9588**	0.9586	0.9585	0.9586	0.9435	0.9567
OVA_Uterus	0.7991	**0.8063**	0.8035	0.8014	0.7412	0.7530
OVA_Kidney	**0.9756**	0.9741	0.9748	0.9755	0.9631	0.9738
OVA_Lung	0.7991	**0.8063**	0.8035	0.7982	0.7412	0.7530
OVA_Omentum	0.7079	0.7072	**0.7095**	0.7053	0.5730	0.6228
OVA_Endometrium	0.7070	0.7179	0.7174	**0.7189**	0.5470	0.6122
OVA_Prostate	**0.9885**	0.9846	0.9845	0.9845	0.9682	0.9688
jannis	0.5160	0.5322	0.5336	0.4914	0.5096	**0.5673**
SVHN	0.8277	0.8402	0.8406	0.7739	0.8245	**0.8736**

Table 7 reports the standard deviation of F1(Macro) scores across 25 repeated runs for each ablation variant.

G Implementation Details of NADT

NADT is implemented in PyTorch and comprises three components: a feature projector, a neighborhood context module (NCM), and a soft decision tree ensemble. This section describes the implementation of each component and the default training configuration.

Feature Projector. The feature projector is a single linear layer `nn.Linear(`d_{in}`, D)` that maps the original d_{in}-dimensional input to a D-dimensional embedding space. The embedding dimension D is set to 64 by default and is automatically increased to 128 when $d_{\text{in}} > 100$, mitigating distance concentration in high-dimensional spaces. The projected embedding is shared between the NCM and the tree ensemble.

Neighborhood Context Module (NCM). The NCM maintains M learnable prototypes as an `nn.Parameter` of shape (M, D), initialized from $\mathcal{N}(0, 0.1^2)$. Before training, prototypes are initialized via MiniBatchKMeans ($n_{\text{init}} = 3$) on the projected training data; if the number of returned clusters is smaller than M, the remaining slots are filled with random noise. A learnable

Table 7: Standard deviation of F1(Macro) scores for NADT and its ablation variants across 30 datasets (5×5 cross-validation).

Dataset	NADT	w/o NCM	w/ Rand $\boldsymbol{\delta}$	w/o Proj.	Axis-Aligned	MLP Router
thyroid	0.0329	0.0000	0.0000	0.0480	0.0347	0.0457
cortex_nuclear	0.0047	0.0285	0.0316	0.0050	0.0041	0.1028
optdigits	0.0035	0.0057	0.0058	0.0037	0.0044	0.0044
texture	0.0015	0.0076	0.0084	0.0025	0.0013	0.0318
banknote-authentication	0.0000	0.0111	0.0114	0.0014	0.0000	0.0000
sonar	0.0452	0.0645	0.0662	0.0605	0.0441	0.0729
pollen	0.0162	0.0160	0.0138	0.0178	0.0186	0.0158
elevators	0.0074	0.0062	0.0063	0.0071	0.0064	0.0067
JapaneseVowels	0.0030	0.0056	0.0055	0.0028	0.0029	0.0037
AP_Colon_Kidney	0.0134	0.0148	0.0131	0.0128	0.0134	0.0137
AP_Endometrium_Prostate	0.0129	0.0107	0.0112	0.0119	0.0227	0.0187
AP_Breast_Omentum	0.0270	0.0281	0.0259	0.0285	0.0316	0.0280
AP_Omentum_Kidney	0.0276	0.0267	0.0260	0.0247	0.0303	0.0277
AP_Omentum_Lung	0.0404	0.0406	0.0401	0.0395	0.0613	0.0385
AP_Uterus_Kidney	0.0206	0.0192	0.0199	0.0197	0.0267	0.0144
AP_Endometrium_Ovary	0.0472	0.0499	0.0553	0.0518	0.0639	0.0461
hiva_agnostic	0.0358	0.0332	0.0369	0.0414	0.0302	0.0628
eating	0.0344	0.0291	0.0258	0.0331	0.0276	0.0382
tumors_C	0.1704	0.1429	0.1505	0.1567	0.1825	0.1439
ovarianTumour	0.1064	0.1155	0.1151	0.1111	0.0446	0.1078
BurkittLymphoma	0.0465	0.0442	0.0420	0.0509	0.0775	0.0587
OVA_Breast	0.0137	0.0135	0.0133	0.0142	0.0156	0.0174
OVA_Uterus	0.0392	0.0370	0.0354	0.0425	0.0520	0.0433
OVA_Kidney	0.0081	0.0081	0.0087	0.0084	0.0125	0.0099
OVA_Lung	0.0392	0.0370	0.0354	0.0427	0.0520	0.0433
OVA_Omentum	0.0704	0.0674	0.0647	0.0761	0.0901	0.0845
OVA_Endometrium	0.0666	0.0651	0.0626	0.0597	0.0928	0.0725
OVA_Prostate	0.0096	0.0131	0.0129	0.0129	0.0290	0.0292
jannis	0.0065	0.0033	0.0032	0.0052	0.0057	0.0063
SVHN	0.0022	0.0023	0.0025	0.0028	0.0025	0.0022

temperature τ is parameterized in log-space as `log_tau`, initialized to $\log\sqrt{D}$ to ensure positivity. The squared L2 distance between each projected sample and prototype is computed via the expanded form $\|\mathbf{x}-\mathbf{p}\|^2 = \|\mathbf{x}\|^2 + \|\mathbf{p}\|^2 - 2\mathbf{x}^\top\mathbf{p}$, which requires only a single matrix multiplication. Affinity weights are then obtained as $\mathbf{w} = \mathrm{Softmax}(-\mathbf{d}^2/\tau)$. A context projection layer `nn.Linear(`M`, `$T \times I$`, bias=False)` maps the M-dimensional affinity vector to a $T \times I$ node-level bias $\boldsymbol{\delta}$, with weights initialized from $\mathcal{N}(0, 0.01^2)$. A LayerNorm is applied to $\boldsymbol{\delta}$ to stabilize its scale. The diversity loss randomly samples 64 prototype pairs and computes a margin hinge loss $\mathcal{L}_{\text{div}} = \mathrm{ReLU}(m - \|\mathbf{p}_a - \mathbf{p}_b\|_2)$ with $m = 2.0$.

Soft Decision Tree Ensemble. Each of the T trees has depth d, yielding $I = 2^d - 1$ internal nodes and $L = 2^d$ leaf nodes. The split weights `split_w` of shape (T, I, F_{sub}) are initialized with Xavier uniform distribution. Split biases `split_b` of shape (T, I) are initialized to zero. Leaf values `leaf_v` of shape (T, L, C) are initialized from $\mathcal{N}(0, 0.01^2)$. Each tree operates on a fixed feature subset of size $F_{\text{sub}} = \max(4, \lfloor d_{\text{in}} \times r \rfloor)$, randomly sampled at initialization and held fixed throughout training. The routing activation at internal node i of tree t is $\text{raw} = \mathbf{w}_{t,i} \cdot \mathbf{x}_{\text{sub}} + b_{t,i} + \delta_{t,i}$, and the routing probability is $p = (\mathrm{softsign}(\text{raw}) + 1)/2$. The path indicator matrix is precomputed via BFS, and the leaf reach probability is the product of routing probabilities along the path. The ensemble prediction averages the logits across all T trees. Dropout is applied to the leaf reach probabilities with a default rate of 0.

Training Protocol. NADT is trained with AdamW using two parameter groups. Base parameters (projector, tree parameters, context projection) use a learning rate of 10^{-3} with weight decay 10^{-4}. Prototype parameters (prototypes and temperature) use a higher learning rate of 5×10^{-3} with no weight decay, encouraging faster adaptation of the neighborhood structure. A cosine annealing scheduler with warm restarts is applied ($T_0 = 50$, $T_{\text{mult}} = 2$, $\eta_{\min} = 10^{-6}$). Gradient norms are clipped to a maximum of 5.0. Label smoothing of 0.1 is applied to the cross-entropy loss. The model is trained for 300 epochs with a batch size of 512.

Default Hyperparameters. Table 8 summarizes the default hyperparameters of NADT. These values are used in all ablation experiments and hyperparameters sensitivity analysis.

Table 8: Default hyperparameters of NADT.

Category	Hyperparameter	Default
Model	Number of trees (T)	128
	Tree depth (d)	4
	Number of prototypes (M)	256
	Feature sampling rate (r)	0.5
	Diversity weight (λ_{div})	0.001
	Diversity margin (m)	2.0
	Label smoothing	0.1
Training	Learning rate (base)	10^{-3}
	Learning rate (prototype)	5×10^{-3}
	Weight decay	10^{-4}
	Batch size	512
	Epochs	300

H Hyperparameter Sensitivity Analysis

We evaluate NADT's sensitivity to five key hyperparameters on all 30 datasets. Each experiment varies a single hyperparameter while keeping others at their default values, using the same evaluation protocol as the main experiments.

Tree depth (depth). Depth controls the expressive capacity of each soft decision tree, directly determining the number of leaf nodes $L = 2^{\text{depth}}$ and internal nodes $I = 2^{\text{depth}} - 1$. Deeper trees capture finer decision boundaries but increase the risk of overfitting, and the multiplicative product of routing probabilities along deep paths exacerbates gradient vanishing. The default depth is 4 (16 leaves, 15 internal nodes), searched over $\{2, 3, 5, 6, 7\}$. Shallow trees (depth $= 2$) suffice for simple tasks but underfit on complex datasets such as BurkittLymphoma. As depth increases, most datasets improve and plateau. However, excessive depth causes degradation on some datasets, tumors_C and eating exhibit monotonic performance decline with depth, reflecting gradient vanishing from the multiplicative routing probabilities. Conversely, large-scale datasets such as SVHN continue to improve with deeper trees, indicating that the optimal depth depends on dataset complexity.

Number of prototypes (M). M determines the clustering granularity of NCM. Larger M captures finer local neighborhood structures for more discriminative node biases, at the cost of increased parameters in the prototype matrix and context projection $W_{\text{ctx}} \in \mathbb{R}^{(T \cdot I) \times M}$. The default $M = 256$, searched over $\{64, 128, 512, 1024\}$. When M is too small, prototypes fail to cover data diversity, reducing bias discriminability. When M is too large, affinity weights approach a uniform distribution, diluting the neighborhood signal and causing sharp degradation on datasets such as jannis and elevators. The preferred M depends on dataset characteristics, as different distributions benefit from different clustering granularities.

Number of trees (T). T controls the number of trees in the ensemble. Each tree independently samples a feature subset and learns its own split parameters and leaf values, with the final prediction obtained by averaging logits across all trees. Increasing T improves ensemble diversity and prediction stability but linearly increases parameter count and training time. The default $T = 128$, searched over $\{16, 32, 64, 256, 512\}$. Performance improves rapidly from $T = 16$ to $T = 64$ and then saturates, with further increases yielding diminishing returns. This saturation indicates that the diversity gain from additional trees is diluted by the averaging effect once the ensemble reaches sufficient size. Large-scale datasets such as SVHN obtain larger gains from larger ensembles, suggesting that complex tasks benefit from greater ensemble capacity.

Diversity loss weight (λ_{div}). λ_{div} controls the strength of prototype diversity regularization. The hinge loss penalizes prototypes whose pairwise distance falls below a margin m, preventing collapse, where prototypes converge to the same location. $\lambda_{\text{div}} = 0$ disables the constraint entirely, while excessively large λ_{div} forces prototypes too far apart, disrupting their alignment with the data

Table 9: F1(Macro) scores under varying tree depth (depth). Best results in bold; second-best underlined.

Dataset	2	3	4	5	6	7
thyroid	0.9042	**0.9598**	0.9436	0.9539	0.9545	0.9541
cortex_nuclear	0.9976	0.9972	0.9969	0.9959	0.9967	**0.9981**
optdigits	0.9809	0.9811	0.9804	0.9816	0.9818	**0.9820**
texture	0.9965	0.9965	**0.9979**	0.9977	0.9975	0.9974
banknote-authentication	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
sonar	**0.8663**	0.8640	0.8655	**0.8663**	0.8611	0.8638
pollen	0.4920	0.4914	0.4935	0.4856	**0.4951**	0.4861
elevators	**0.8547**	0.8515	0.8531	0.8513	0.8510	0.8508
JapaneseVowels	**0.9935**	0.9933	0.9926	0.9934	0.9926	0.9916
AP_Colon_Kidney	0.9789	**0.9807**	0.9791	0.9798	0.9789	0.9798
AP_Endo._Prostate	0.9923	**0.9961**	0.9953	0.9923	0.9884	0.9923
AP_Breast_Omentum	0.9345	**0.9482**	0.9405	0.9369	0.9330	0.9373
AP_Omentum_Kidney	0.9640	0.9615	0.9635	**0.9652**	0.9631	0.9633
AP_Omentum_Lung	0.9194	0.9193	0.9155	0.9192	**0.9217**	0.9166
AP_Uterus_Kidney	**0.9750**	0.9720	0.9725	0.9719	0.9733	0.9719
AP_Endo._Ovary	0.8861	0.8889	0.8900	0.8891	**0.8934**	0.8836
hiva_agnostic	0.6891	0.6938	0.6852	**0.6942**	0.6830	0.6780
eating	**0.6492**	0.6479	0.6371	0.6424	0.6262	0.6204
tumors_C	**0.6181**	0.5786	0.5649	0.4805	0.5206	0.5523
ovarianTumour	0.6669	**0.6924**	0.6796	0.6673	0.6489	0.6660
BurkittLymphoma	0.8017	0.8310	0.8545	**0.8589**	0.8409	0.8282
OVA_Breast	0.9571	0.9596	0.9588	0.9592	**0.9605**	0.9591
OVA_Uterus	**0.8027**	0.7949	0.7991	0.7905	0.7704	0.7601
OVA_Kidney	0.9755	**0.9759**	0.9756	0.9746	0.9745	0.9750
OVA_Lung	**0.8027**	0.7949	0.7991	0.7905	0.7704	0.7601
OVA_Omentum	0.6989	0.6999	0.7079	0.6957	**0.7108**	0.6876
OVA_Endometrium	0.6689	0.7035	0.7070	**0.7222**	0.7189	0.6996
OVA_Prostate	**0.9885**	**0.9885**	**0.9885**	0.9864	0.9842	0.9842
jannis	0.5059	0.5119	0.5160	0.5177	0.5182	**0.5236**
SVHN	0.8120	0.8206	0.8277	0.8323	0.8340	**0.8368**

distribution. The default $\lambda_{\text{div}} = 0.001$, searched over $\{0, 10^{-4}, 10^{-2}, 10^{-1}\}$. Removing the diversity loss causes clear degradation on some datasets, confirming that prototype collapse harms bias discriminability. A few datasets perform best without regularization, indicating that its necessity depends on the data distribution. Overall, NADT is robust to λ_{div}, with limited performance fluctuation across values.

Feature sampling ratio (r). r controls the fraction of features sampled by each tree at initialization, with $F_{\text{sub}} = \lfloor d \times r \rfloor$ feature indices fixed throughout training. Smaller r increases inter-tree feature diversity for better ensemble generalization, while excessively small r may starve individual trees of information. The default $r = 0.5$, searched over $\{0.3, 0.7, 0.9, 1.0\}$. Performance is largely insensitive to r, with most datasets showing limited variation across values, demonstrating strong robustness. On small-sample tasks, larger r tends to improve performance by retaining more features for individual tree learning, while on simple tasks smaller r gains an edge through greater tree diversity.

I Runtime Analysis

All experiments are conducted on the same hardware (NVIDIA GPU with CUDA support). Table 14 reports the total training time including Optuna hyperparameter tuning and 5×5 cross-validation. KNDT's training time lies between highly optimized traditional libraries and computationally expensive deep learning methods. On SVHN, KNDT (595s) is comparable to ModernNCA (532s) but dramatically faster than TabR (18,323s), GRANDE (3,456s), and SoHoT (42,349s). SoHoT exhibits the most severe bottleneck, exceeding 1,000s on six datasets. Random Forest is consistently the fastest, while CatBoost is notably slow on high-dimensional datasets due to ordered boosting overhead. This favorable scaling stems from KNDT's architectural design: NCM performs fixed-

Table 10: F1(Macro) scores under varying number of prototypes (M). Best results in bold; second-best underlined.

Dataset	64	128	256	512	1024
thyroid	0.9523	0.9573	0.9436	0.9536	**0.9574**
cortex_nuclear	0.9944	0.9963	0.9969	**0.9973**	0.9958
optdigits	0.9792	0.9798	0.9804	0.9839	**0.9842**
texture	0.9969	0.9972	**0.9979**	0.9976	0.9970
banknote-authentication	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
sonar	0.8515	0.8612	0.8655	0.8735	**0.8785**
pollen	0.4923	0.4897	**0.4935**	0.4821	0.4921
elevators	**0.8769**	0.8652	0.8531	0.8397	0.8253
JapaneseVowels	0.9890	0.9920	0.9926	0.9937	**0.9946**
AP_Colon_Kidney	**0.9807**	0.9789	0.9791	0.9798	0.9798
AP_Endo._Prostate	**0.9961**	**0.9961**	0.9953	**0.9961**	**0.9961**
AP_Breast_Omentum	0.9379	**0.9430**	0.9405	0.9417	0.9397
AP_Omentum_Kidney	0.9615	0.9634	0.9635	0.9615	**0.9652**
AP_Omentum_Lung	**0.9218**	0.9167	0.9155	**0.9218**	**0.9218**
AP_Uterus_Kidney	0.9749	**0.9763**	0.9725	0.9733	0.9750
AP_Endo._Ovary	0.8900	0.8862	0.8900	0.8841	**0.8928**
hiva_agnostic	**0.7017**	0.6895	0.6852	0.6895	0.6954
eating	**0.6506**	0.6472	0.6371	0.6362	0.6364
tumors_C	0.5856	0.5734	0.5649	0.5864	**0.6179**
ovarianTumour	0.6774	0.6671	**0.6796**	0.6670	0.6581
BurkittLymphoma	0.8642	0.8503	0.8545	0.8499	**0.8806**
OVA_Breast	**0.9617**	0.9613	0.9588	0.9602	**0.9617**
OVA_Uterus	0.7943	0.7927	0.7991	0.7851	**0.7995**
OVA_Kidney	**0.9770**	**0.9770**	0.9756	0.9758	0.9758
OVA_Lung	0.7943	0.7927	0.7991	0.7851	**0.7995**
OVA_Omentum	0.7013	**0.7229**	0.7079	0.6990	0.7048
OVA_Endometrium	**0.7147**	0.7083	0.7070	0.6937	0.6973
OVA_Prostate	**0.9885**	0.9864	**0.9885**	0.9864	**0.9885**
jannis	**0.5595**	0.5346	0.5160	0.5027	0.4928
SVHN	**0.8362**	0.8328	0.8277	0.8274	0.8310

cost prototype distance computation, and the tree ensemble operates on compact bias vectors rather than full feature representations.

Table 15 reports per-fold inference time. KNDT achieves sub-0.1s inference on 29 of 30 datasets, with only SVHN (0.621s) slightly exceeding this threshold. This is orders of magnitude faster than ProtoGate (up to 389s on OVA_Breast), GRANDE (up to 76s on ovarianTumour), TabR (37s on SVHN), and SoHoT (32s on jannis). KNDT's inference efficiency comes from its modular design: NCM produces compact node bias vectors via fixed-cost computation, and tree routing involves only a forward pass with cost proportional to $T\times$ depth, without training-set retrieval. ModernNCA achieves comparable inference speed through a simple MLP forward pass.

Table 11: F1(Macro) scores under varying number of trees (T). Best results in bold; second-best underlined.

Dataset	16	32	64	128	256	512
thyroid	0.9489	0.9517	0.9539	0.9436	**0.9567**	0.9539
cortex_nuclear	0.9967	0.9967	**0.9972**	0.9969	**0.9972**	0.9963
optdigits	**0.9817**	0.9802	0.9806	0.9804	0.9804	0.9807
texture	0.9973	0.9977	0.9972	**0.9979**	0.9975	0.9972
banknote-authentication	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
sonar	0.8666	0.8666	0.8637	0.8655	**0.8736**	0.8711
pollen	0.4866	0.4948	0.4867	0.4935	0.4964	**0.4965**
elevators	0.8536	**0.8567**	0.8533	0.8531	0.8537	0.8532
JapaneseVowels	0.9928	0.9924	0.9930	0.9926	0.9931	**0.9936**
AP_Colon_Kidney	0.9798	0.9789	0.9780	0.9791	0.9798	**0.9807**
AP_Endo._Prostate	**0.9961**	**0.9961**	0.9923	0.9953	0.9884	0.9923
AP_Breast_Omentum	**0.9415**	0.9397	0.9394	0.9405	0.9397	0.9413
AP_Omentum_Kidney	0.9594	0.9615	0.9633	0.9635	0.9633	**0.9652**
AP_Omentum_Lung	0.9216	0.9193	0.9166	0.9155	**0.9243**	0.9192
AP_Uterus_Kidney	0.9735	0.9749	0.9749	0.9725	0.9734	**0.9750**
AP_Endo._Ovary	0.8874	**0.8925**	0.8901	0.8900	0.8878	0.8873
hiva_agnostic	0.6787	0.6903	0.6915	0.6852	0.6878	**0.6965**
eating	0.6412	0.6434	0.6525	0.6371	0.6475	**0.6567**
tumors_C	**0.6325**	0.5859	0.5971	0.5649	0.5898	0.5676
ovarianTumour	0.6631	**0.6810**	0.6798	0.6796	**0.6810**	0.6638
BurkittLymphoma	0.8611	**0.8706**	0.8678	0.8545	0.8542	0.8456
OVA_Breast	0.9597	0.9607	0.9607	0.9588	0.9608	**0.9616**
OVA_Uterus	0.7829	0.7960	0.7892	0.7991	**0.8005**	0.7879
OVA_Kidney	0.9751	0.9758	**0.9770**	0.9756	0.9764	0.9753
OVA_Lung	0.7829	0.7960	0.7892	0.7991	**0.8005**	0.7879
OVA_Omentum	0.7062	0.7072	0.7048	**0.7079**	0.7069	0.6982
OVA_Endometrium	0.6848	0.6966	0.7000	0.7070	**0.7187**	0.7023
OVA_Prostate	0.9864	0.9862	0.9864	**0.9885**	0.9864	**0.9885**
jannis	0.5095	0.5107	0.5135	0.5160	**0.5178**	0.5147
SVHN	0.8202	0.8247	0.8267	0.8277	0.8281	**0.8285**

Table 12: F1(Macro) scores under varying diversity loss weight (λ_{div}). Best results in bold; second-best underlined.

Dataset	0	10^{-4}	10^{-3}	10^{-2}	10^{-1}
thyroid	**0.9584**	**0.9584**	0.9436	0.9545	0.9567
cortex_nuclear	0.9968	0.9968	0.9969	**0.9972**	0.9968
optdigits	0.9807	0.9809	0.9804	0.9814	**0.9816**
texture	0.9976	0.9972	**0.9979**	0.9974	0.9976
banknote-authentication	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
sonar	0.8590	0.8615	0.8655	0.8663	**0.8712**
pollen	0.4928	0.4900	0.4935	0.4874	**0.4944**
elevators	0.8525	0.8515	**0.8531**	0.8494	0.8502
JapaneseVowels	0.9928	0.9928	0.9926	0.9930	**0.9934**
AP_Colon_Kidney	**0.9798**	**0.9798**	0.9791	**0.9798**	**0.9798**
AP_Endo._Prostate	0.9923	0.9923	**0.9953**	0.9923	0.9923
AP_Breast_Omentum	0.9397	0.9397	**0.9405**	0.9397	0.9397
AP_Omentum_Kidney	0.9615	0.9615	**0.9635**	0.9615	0.9615
AP_Omentum_Lung	**0.9243**	**0.9243**	0.9155	0.9218	**0.9243**
AP_Uterus_Kidney	0.9735	0.9720	0.9725	0.9720	**0.9750**
AP_Endo._Ovary	0.8841	0.8869	**0.8900**	0.8862	0.8820
hiva_agnostic	0.6818	0.6864	0.6852	0.6869	**0.6905**
eating	0.6483	**0.6498**	0.6371	0.6407	0.6417
tumors_C	0.5624	**0.6006**	0.5649	0.5805	0.5805
ovarianTumour	0.6581	0.6581	**0.6796**	0.6581	0.6581
BurkittLymphoma	0.8564	0.8522	0.8545	**0.8679**	0.8669
OVA_Breast	0.9607	0.9607	0.9588	0.9617	**0.9621**
OVA_Uterus	0.7928	0.7912	**0.7991**	0.7889	0.7918
OVA_Kidney	**0.9764**	**0.9764**	0.9756	**0.9764**	**0.9764**
OVA_Lung	0.7928	0.7912	**0.7991**	0.7889	0.7918
OVA_Omentum	0.7032	0.6997	**0.7079**	0.7045	0.7040
OVA_Endometrium	0.7064	0.7128	0.7070	**0.7145**	0.6990
OVA_Prostate	**0.9885**	**0.9885**	**0.9885**	**0.9885**	**0.9885**
jannis	0.5132	0.5152	0.5160	**0.5181**	0.5170
SVHN	**0.8277**	0.8270	**0.8277**	0.8271	0.8269

Table 13: F1(Macro) scores under varying feature sampling ratio (r). Best results in bold; second-best underlined.

Dataset	0.3	0.5	0.7	0.9	1.0
thyroid	**0.9539**	0.9436	**0.9539**	**0.9539**	**0.9539**
cortex_nuclear	0.9967	**0.9969**	0.9963	0.9963	0.9964
optdigits	0.9801	0.9804	**0.9813**	0.9806	0.9804
texture	0.9971	**0.9979**	0.9976	0.9969	0.9972
banknote-authentication	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
sonar	**0.8711**	0.8655	0.8685	0.8685	0.8634
pollen	0.4894	**0.4935**	0.4894	0.4894	0.4906
elevators	0.8504	0.8531	**0.8548**	0.8528	0.8535
JapaneseVowels	0.9926	0.9926	**0.9947**	0.9926	0.9925
AP_Colon_Kidney	**0.9798**	0.9791	**0.9798**	**0.9798**	**0.9798**
AP_Endo._Prostate	0.9923	**0.9953**	0.9884	0.9923	0.9923
AP_Breast_Omentum	0.9378	**0.9405**	0.9360	0.9379	0.9379
AP_Omentum_Kidney	0.9615	**0.9635**	0.9633	0.9634	0.9615
AP_Omentum_Lung	0.9216	0.9155	0.9193	0.9167	**0.9217**
AP_Uterus_Kidney	0.9705	0.9725	0.9734	**0.9764**	**0.9764**
AP_Endo._Ovary	0.8844	0.8900	**0.9015**	0.8942	0.8963
hiva_agnostic	0.6759	0.6852	**0.6973**	0.6911	0.6896
eating	**0.6541**	0.6371	0.6487	0.6469	0.6502
tumors_C	0.5805	0.5649	0.6018	**0.6169**	0.5988
ovarianTumour	0.6692	0.6796	**0.6810**	0.6626	0.6581
BurkittLymphoma	0.8465	0.8545	0.8700	**0.8795**	0.8639
OVA_Breast	0.9594	0.9588	0.9607	0.9612	**0.9622**
OVA_Uterus	0.7947	0.7991	0.7854	0.7948	**0.8011**
OVA_Kidney	0.9759	0.9756	0.9759	0.9748	**0.9764**
OVA_Lung	0.7947	0.7991	0.7854	0.7948	**0.8011**
OVA_Omentum	0.7165	0.7079	0.7026	0.7130	**0.7177**
OVA_Endometrium	0.6691	0.7070	0.6966	0.6948	**0.7119**
OVA_Prostate	0.9864	**0.9885**	0.9864	0.9864	0.9864
jannis	0.5133	**0.5160**	0.5141	0.5148	0.5154
SVHN	**0.8285**	0.8277	0.8255	0.8239	0.8245

Table 14: Training time (seconds). Lower is better. "—" indicates that the method could not run on the dataset.

Dataset	NADT	RF	XGBoost	LGBM	CatBoost	MNCA	TabR	P.Gate	DTSem	DOFEN	GRANDE	SoHoT
thyroid	1.864	0.055	0.494	1.477	0.976	0.587	0.356	6.429	0.914	0.203	17.633	5.962
cortex_nuclear	6.022	0.276	12.845	23.466	17.358	4.909	1.875	20.235	7.442	5.637	54.93	110.417
optdigits	15.284	0.683	129.503	151.956	9.83	12.283	11.047	—	46.748	27.702	467.929	586.263
texture	15.212	0.629	42.493	105.036	10.759	10.197	16.939	—	39.607	60.004	186.42	603.154
banknote-authentication	4.705	0.763	37.537	64.393	2.43	2.652	2.219	15.029	4.6	4.427	27.311	287.229
sonar	1.535	0.897	17.471	17.338	4.769	0.394	0.419	4.897	0.817	1.777	23.695	6.435
pollen	11.315	0.862	8.124	32.69	1.56	17.667	9.014	—	12.619	1.26	20.313	146.013
elevators	52.443	0.758	30.224	89.727	5.695	43.616	53.086	—	54.561	35.529	514.765	3949.509
JapaneseVowels	26.645	0.594	26.985	61.224	3.426	14.469	27.574	—	33.196	15.244	229.034	1896.694
AP_Colon_Kidney	6.345	0.223	13.994	21.7	88.865	0.578	2.474	40.093	1.764	3.156	44.101	2034.651
AP_Endo._Prostate	1.472	0.196	1.585	1.64	27.573	0.345	0.685	32.796	0.486	1.754	35.248	471.968
AP_Breast_Omentum	2.843	0.263	18.703	32.243	18.268	0.332	2.448	44.773	1.4	2.491	86.408	1642.158
AP_Omentum_Kidney	2.282	0.394	29.947	30.975	62.215	0.304	2.659	36.406	1.14	2.082	68.583	1274.522
AP_Omentum_Lung	1.82	0.836	26.704	22.85	219.555	0.712	0.702	35.729	0.749	2.283	42.328	777.078
AP_Uterus_Kidney	2.002	0.722	32.823	14.644	50.604	0.428	1.853	19.039	1.277	2.281	56.969	1462.902
AP_Endo._Ovary	1.643	0.194	8.482	6.591	38.151	0.242	0.746	29.761	0.899	2.681	45.841	1043.194
hiva_agnostic	12.009	0.352	1.102	4.235	15.022	11.793	20.018	—	14.725	12.988	98.402	1081.344
eating	3.924	0.839	65.536	173.065	295.827	2.808	3.842	30.357	6.366	12.093	70.643	2033.734
tumors_C	1.329	0.522	7.014	5.536	18.483	0.242	0.384	31.274	0.239	1.313	21.246	145.865
ovarianTumour	4.186	0.916	355.178	103.576	244.485	0.348	2.318	225.781	1.418	3.705	206.532	5143.295
BurkittLymphoma	3.114	0.348	70.971	21.762	41.167	0.349	1.016	74.706	0.921	3.219	78.54	1613.225
OVA_Breast	13.967	0.503	32.152	134.573	52.955	4.196	14.223	232.605	4.804	7.1	221.993	5849.152
OVA_Uterus	12.151	2.174	35.094	8.875	48.426	1.553	12.73	229.118	4.709	7.165	111.565	5616.154
OVA_Kidney	9.402	1.281	11.799	151.349	58.357	1.524	12.736	229.393	5.071	7.101	79.838	6170.852
OVA_Lung	6.59	2.435	25.789	62.766	108.914	1.3	13.249	233.974	4.899	7.116	112.605	6846.412
OVA_Omentum	5.526	0.53	54.952	21.887	226.048	1.334	13.647	231.798	4.95	7.131	104.133	5739.327
OVA_Endometrium	9.423	0.498	23.508	10.09	69.366	1.231	12.7	230.476	4.914	7.125	128.434	6246.096
OVA_Prostate	8.584	0.481	22.225	58.599	27.353	1.132	13.212	232.445	5.035	7.12	84.694	6909.694
jannis	214.117	7.494	2.799	4.555	8.417	321.715	712.058	—	385.481	220.962	3012.57	17251.755
SVHN	594.961	47.638	516.088	393.425	1800.87	532.175	18322.705	—	839.503	1901.092	3455.703	42348.709

Table 15: Inference time (seconds). Lower is better. "—" indicates that the method could not run on the dataset.

Dataset	NADT	RF	XGBoost	LGBM	CatBoost	MNCA	TabR	P.Gate	DTSem	DOFEN	GRANDE	SoHoT
thyroid	0.003	0.03	0.005	0.004	0.001	0.002	0.003	0.031	0.003	0.001	0.196	0.02
cortex_nuclear	0.003	0.101	0.004	0.007	0.01	0.002	0.004	1.654	0.005	0.012	0.311	0.081
optdigits	0.005	0.551	0.017	0.018	0.074	0.005	0.034	—	0.023	0.06	0.819	0.769
texture	0.004	0.318	0.011	0.015	0.034	0.005	0.051	—	0.022	0.419	0.76	0.836
banknote-authentication	0.001	0.631	0.023	0.014	0.001	0.001	0.015	2.714	0.005	0.015	0.207	0.359
sonar	0.002	0.798	0.013	0.008	0.002	0.001	0.003	0.009	0.002	0.005	0.229	0.012
pollen	0.003	0.584	0.014	0.011	0.002	0.003	0.033	—	0.01	0.008	0.329	0.38
elevators	0.009	0.405	0.017	0.016	0.103	0.024	0.196	—	0.036	0.066	2.054	6.98
JapaneseVowels	0.005	0.396	0.019	0.013	0.03	0.006	0.079	—	0.02	0.031	0.704	3.7
AP_Colon_Kidney	0.021	0.074	0.021	0.063	0.558	0.003	0.009	1.228	0.005	0.02	9.307	0.045
AP_Endo._Prostate	0.004	0.111	0.014	0.05	0.136	0.002	0.005	0.048	0.002	0.009	9.116	0.011
AP_Breast_Omentum	0.01	0.142	0.034	0.031	0.41	0.003	0.008	0.741	0.004	0.016	9.224	0.031
AP_Omentum_Kidney	0.01	0.13	0.05	0.107	0.698	0.003	0.013	0.469	0.003	0.014	9.568	0.027
AP_Omentum_Lung	0.006	0.54	0.044	0.069	0.328	0.003	0.005	0.194	0.003	0.014	9.174	0.015
AP_Uterus_Kidney	0.01	0.346	0.039	0.025	0.414	0.003	0.006	0.673	0.004	0.015	9.708	0.028
AP_Endo._Ovary	0.007	0.093	0.032	0.062	0.458	0.002	0.005	0.261	0.003	0.017	9.519	0.022
hiva_agnostic	0.014	0.149	0.006	0.008	0.557	0.004	0.041	—	0.016	0.107	1.754	0.603
eating	0.013	0.158	0.027	0.024	0.792	0.004	0.01	3.339	0.006	0.028	5.322	0.076
tumors_C	0.002	0.4	0.022	0.032	0.087	0.001	0.009	0.009	0.002	0.006	5.551	0.009
ovarianTumour	0.034	0.545	0.2	0.374	0.673	0.007	0.016	1.826	0.007	0.019	75.785	0.018
BurkittLymphoma	0.019	0.196	0.088	0.04	0.683	0.005	0.008	0.258	0.004	0.015	22.331	0.012
OVA_Breast	0.054	0.062	0.029	0.068	2.289	0.009	0.039	383.518	0.011	0.058	9.911	0.164
OVA_Uterus	0.049	0.239	0.035	0.051	1.987	0.008	0.034	388.71	0.01	0.057	9.608	0.177
OVA_Kidney	0.032	0.273	0.026	0.105	1.597	0.008	0.034	368.78	0.011	0.057	9.705	0.171
OVA_Lung	0.029	0.538	0.059	0.113	2.005	0.008	0.035	385.925	0.01	0.057	9.614	0.217
OVA_Omentum	0.028	0.084	0.058	0.067	2.904	0.008	0.035	357.498	0.009	0.057	9.67	0.19
OVA_Endometrium	0.032	0.143	0.046	0.041	1.973	0.008	0.034	366.907	0.01	0.057	9.623	0.263
OVA_Prostate	0.031	0.086	0.035	0.076	0.99	0.008	0.036	347.717	0.01	0.057	9.575	0.141
jannis	0.048	0.079	0.011	0.041	0.203	0.253	2.644	—	0.118	0.282	11.434	31.582
SVHN	0.621	0.471	0.161	0.366	16.385	0.416	36.803	—	0.271	4.124	25.978	16.734