LLM-Enhanced Random Forests in Orthogonal Hyperbolic Subspaces for Tabular Learning

Shengpeng Wang[1,2], **Yisen Gao**[3], **Han Xiang**[1,2], **Lingyun Liu**[1,2], **Ziwei Zhang**[4]
Qingyun Sun[4,5], **Xianxian Li**[1,2], **Xingcheng Fu**[1,2*]
[1]Key Lab of Education Blockchain and Intelligent Technology, Guangxi Normal University
[2]Guangxi Key Lab of Multi-source Information Mining & Security, Guangxi Normal University
[3]Computer Science and Engineering, The Hong Kong University of Science and Technology
[4]School of Computer Science and Engineering, Beihang University
[5]Zhejiang Key Laboratory of Intelligent Education Technology and Application,
Zhejiang Normal University, Jinhua, Zhejiang, China
SPalmW@stu.gxnu.edu.cn, {zwzhang,sunqy}@buaa.edu.cn, fuxc@gxnu.edu.cn

Abstract

Tabular data is widely applied in critical domains, where tree-based models remain dominant, yet their performance is often constrained by manual heuristic splitting criteria and limited ensemble diversity. The hierarchical partitions induced by decision trees motivate geometric priors for guiding splits and constructing diverse ensembles. Hyperbolic geometry naturally accommodates such hierarchical structures, but existing hyperbolic algorithms are hindered by complex Riemannian optimization and the difficulty of defining subtree priors. To address these challenges, we propose HOT, a hyperbolic random forest framework based on data decoupling. First, we leverage intrinsic geometric structures and LLM reasoning to guide subtree construction, replacing traditional manual heuristics with geometric-semantic priors. Second, we introduce an orthogonal hyperbolic subspace projection mechanism via tangent space isometry, effectively decoupling feature dependencies and maximizing ensemble diversity. Finally, we establish an end-to-end collaborative training paradigm where gradient residuals from a graph network directly guide the growth of new hyperbolic trees, bypassing expensive iterative optimization. Experiments on 13 benchmark datasets demonstrate that HOT significantly outperforms state-of-the-art baselines in both computational efficiency and predictive performance.

1 Introduction

Tabular data is ubiquitous in the world and is stored in relational datasets in a structured format. It is widely applied in critical domains such as financial risk control [2], medical diagnosis [22], and advertising [15]. Given the inherent heterogeneity of tabular features, tree-based methods have consistently proven to be highly effective for processing tabular data. The superiority of tree ensemble models mainly comes from their strong ability to handle feature heterogeneity and learn irregular, piecewise constant target functions.

To explore further possibilities for tabular data, recent works mainly focus on using deep learning (DL) [4], Graph Neural Networks (GNNs) [31], and Large Language Models (LLMs) [20] to enhance representation capabilities. DL-based methods use gradient-based optimization to refine decision boundaries, while GNN-based methods aim to explicitly capture topological correlations among samples. Additionally, LLMs leverage semantic reasoning to optimize decision rules or enhance tree

*Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

diversity. However, these methods fundamentally struggle to bridge the gap between discrete logic and continuous geometry. Although they achieve promising results, the performance ceiling of tree ensemble models is consistently limited by the diversity among individual trees [6].

On the other hand, hyperbolic geometry has attracted widespread attention in the machine learning field [16, 36] due to its unique curvature and property that its spatial volume grows exponentially with the radius. This geometric property strictly matches the exponential expansion of nodes when decision trees recursively partition data. As tree depth increases, sample categories are continuously subdivided. The limited capacity of Euclidean space leads to overlap and representation distortion for deep level data points. In contrast, with its exponentially growing spatial capacity, hyperbolic space provides independent and non-interfering representation regions for every branch of a decision tree [11]. This natural alignment between spatial capacity and tree-like data topology effectively guarantees the geometric fidelity of hierarchical features. Therefore, mapping data into a hyperbolic latent space is an ideal approach for building efficient tree models.

Therefore, a natural idea is to generate the decision tree [38] directly from the representation of the data in the latent space based on hyperbolic geometry, which can effectively capture the latent semantic associations and the underlying hierarchical structure. However, this method still faces the following two core **challenges**: (1)Defining effective prior distributions and generating subtrees under non-Euclidean constraints [26] remains an open problem, as traditional manual heuristics are difficult to generalize to curved manifolds. (2)Existing hyperbolic algorithms heavily rely on Riemannian optimization, which suffers from prohibitive computational costs and struggles to achieve the structural diversity necessary for high-dimensional, heterogeneous tabular data.

To address the aforementioned challenges, we propose a random forest construction method based on data decoupling in hyperbolic space. Firstly, we propose a mechanism that integrates the generation of hyperbolic decision trees with the reasoning capabilities of large language models (LLMs). This mechanism leverages the intrinsic geometric structure of the data and the reasoning power of LLMs to guide the construction of subtrees, rather than relying on traditional manual heuristic rules. Second, we introduce orthogonal hyperbolic subspace projection, which achieves effective decoupling of feature dependencies and maximizes diversity between subtrees by generating them in orthogonal subspace, thereby significantly improving the ensemble performance of the random forest while ensuring individual quality. Finally, to alleviate the challenge of high Riemannian optimization costs, we embed tabular data into a hyperbolic latent space. We utilize the geometric properties of this space to achieve generation of decision trees. Furthermore, we use LLMs to capture the latent semantic hierarchical structures within the tabular data. By employing a gradient-driven residual fitting mechanism, we use these semantic rules as structural priors to guide the error correction of the graph network, thereby achieving an end-to-end training mode. This synergy avoids redundant iterative optimization processes and effectively resolves the efficiency bottleneck in processing large-scale tabular data. We highlight the advantage of HOT as follows:

- Unlike traditional methods that rely on manual heuristic rules, we propose a data decoupling mechanism that leverages the intrinsic hyperbolic geometric structure of the data. By using geometric priors to guide subtree construction, this approach solves the challenge of defining prior distributions for subtrees in non-Euclidean spaces.
- Based on the isometric isomorphism of the tangent space, we introduce an orthogonal hyperbolic subspace projection strategy. This achieves the effective decoupling of feature dependencies and maximizes the variance among subtrees.
- We conducted experiments on various benchmarks, and the results demonstrated the superiority of our method.

2 Preliminary

2.1 Problem Definition

Given a tabular dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$ containing N samples. Tabular data inherently consists of heterogeneous features, encompassing both continuous numerical features and discrete categorical features. Let the raw feature space be $\mathcal{X}_i^{raw} = (\mathcal{X}_i^{num}, \mathcal{X}_i^{cat})$. Following standard tabular preprocessing protocols, categorical features are mapped to dense embeddings (or one-hot encoded) and concatenated with normalized numerical features to form a unified continuous initial representation

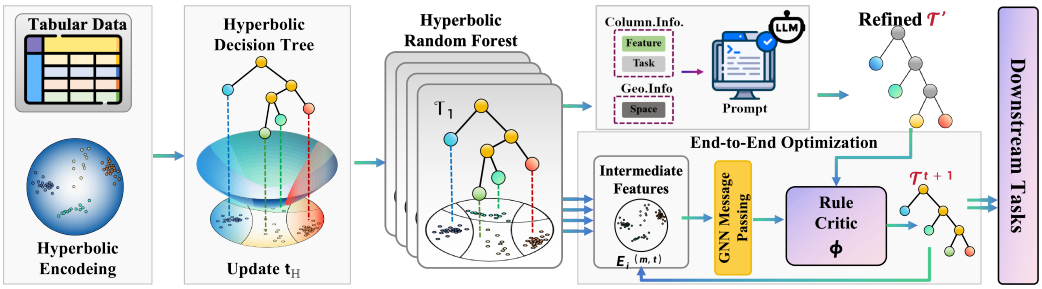


Figure 1: Overview of HOT. (1) Orthogonal hyperbolic forest construction maps tabular features into orthogonal hyperbolic subspaces and constructs diverse trees with geometrically corrected splits. (2) LLM-based rule refinement uses task and geometric information to refine decision rules. (3) End-to-end alternating optimization uses gradient residuals from a graph network to guide new tree fitting and iteratively refine representations for downstream prediction.

$\mathbf{x}_i \in \mathbb{R}^d$. The corresponding label is $y_i \in \mathcal{Y}$. For classification tasks, $\mathcal{Y} = \{1, \ldots, K\}$; for regression tasks, $\mathcal{Y} = \mathbb{R}$. The goal is to construct a mapping function $F : \mathbb{R}^d \to \mathcal{Y}$, which consists of a forest $\mathcal{F} = \{T_1, \ldots, T_M\}$ composed of a set of decision trees.

2.2 Hyperbolic Space

To effectively embed tabular data with implicit hierarchical structures, we adopt the Lorentz model in hyperbolic geometry. Let $\mathbb{R}^{D+1}$ be a vector space equipped with the Minkowski inner product $\langle \cdot, \cdot \rangle_{\mathcal{L}}$. For any two vectors $\mathbf{u}, \mathbf{v} \in \mathbb{R}^{D+1}$, their inner product is defined as:

$$\langle \mathbf{u}, \mathbf{v} \rangle_{\mathcal{L}} = -u_0 v_0 + \sum_{j=1}^{D} u_j v_j, \tag{1}$$

where u_0, v_0 are the time components, and $u_{1:D}, v_{1:D}$ are the spatial components.

Based on this, the D-dimensional Lorentz manifold $\mathbb{H}^D_\kappa$ with curvature $-\kappa$ ($\kappa > 0$) is defined as:

$$\mathbb{H}^D_\kappa = \{\mathbf{z} \in \mathbb{R}^{D+1} \mid \langle \mathbf{z}, \mathbf{z} \rangle_{\mathcal{L}} = -1/\kappa, z_0 > 0\}, \tag{2}$$

where the time component z_0 of any point $\mathbf{z}$ is uniquely determined by its spatial components, specifically $z_0 = \sqrt{1/\kappa + \|\mathbf{z}_{1:D}\|_2^2}$. At the origin $\mathbf{o} = (\sqrt{1/\kappa}, \mathbf{0})^\top$ of the Lorentz model, the tangent space $\mathcal{T}_{\mathbf{o}}\mathbb{H}^D_\kappa$ approximates Euclidean space. For any tangent vector $\mathbf{v} = (0, \bar{\mathbf{v}})^\top \in \mathcal{T}_{\mathbf{o}}\mathbb{H}^D_\kappa$, it satisfies $\langle \mathbf{o}, \mathbf{v} \rangle_{\mathcal{L}} = 0$.

The exponential map $\exp_{\mathbf{o}} : \mathcal{T}_{\mathbf{o}}\mathbb{H}^D_\kappa \to \mathbb{H}^D_\kappa$ projects tangent space vectors back onto the manifold; the logarithmic map $\log_{\mathbf{o}} : \mathbb{H}^D_\kappa \to \mathcal{T}_{\mathbf{o}}\mathbb{H}^D_\kappa$ is used to map points on the manifold back to the tangent space to perform Euclidean operations. Their definitions are:

$$\exp_{\mathbf{o}}(\mathbf{v}) = \cosh(\|\mathbf{v}\|_{\mathcal{L}})\mathbf{o} + \sinh(\|\mathbf{v}\|_{\mathcal{L}})\frac{\mathbf{v}}{\|\mathbf{v}\|_{\mathcal{L}}}, \tag{3}$$

$$\log_{\mathbf{o}}(\mathbf{z}) = \frac{\operatorname{arccosh}(-\kappa\langle \mathbf{o}, \mathbf{z} \rangle_{\mathcal{L}})}{\sqrt{\langle \mathbf{o}, \mathbf{z} \rangle_{\mathcal{L}}^2 - 1}}(\mathbf{z} + \langle \mathbf{o}, \mathbf{z} \rangle_{\mathcal{L}}\mathbf{o}), \tag{4}$$

where $\|\mathbf{v}\|_{\mathcal{L}} = \sqrt{\langle \mathbf{v}, \mathbf{v} \rangle_{\mathcal{L}}}$ represents the norm of the tangent vector under the Minkowski metric (for a tangent vector at the origin, this is equivalent to the Euclidean norm $\|\bar{\mathbf{v}}\|_2$).

3 Method

In this section, our HOT overall framework is shown in Figure 1, which aims to effectively capture the implicit complex hierarchical topological structures in tabular data through deep integration of geometric priors and semantic reasoning. By mapping tabular data into hyperbolic space, we construct a latent space with intrinsic hierarchical properties. Then, we project the global high-dimensional

manifold into mutually independent low-dimensional hyperbolic subspaces to achieve diversity among these subspaces. Afterwards, we utilize LLMs to semantically refine the rules extracted from the forest, generating a refined rule set. Subsequently, we introduce an end-to-end alternating optimization mechanism based on gradient residuals, thereby achieving a deep co-evolution of continuous graph topologies and discrete tabular semantics.

Our aim is to introduce multiple orthogonal hyperbolic subspaces to generate diverse-biased decision tree rules, as illustrated in Figure 2. However, a core challenge is the calculation of orthogonality between two or more curved manifolds (spaces), which incurs high computational costs. Therefore, inspired by differential geometry, we have provided an extended definition and approximation algorithm of random forest generation for orthogonal hyperbolic subspaces based on a unified tangent space.

3.1 Orthogonal Hyperbolic Subspace

Hyperbolic Representation of Tabular Data. To capture the complex hierarchical interactions latent within tabular features, we map the raw data into a high-dimensional Lorentz latent space. Let $\mathbf{X} \in \mathbb{R}^{N\times d}$ be the input data matrix. We construct a Hyperbolic Geometric Encoder $\Phi : \mathbb{R}^d \to \mathbb{H}^D_\kappa$, lifting features from low-dimensional Euclidean space to a high-dimensional hyperbolic manifold $\mathbb{H}^D_\kappa$ (where $D \gg d$ is the latent space dimension). This process utilizes the tangent space at the origin $\mathbf{o} = (\sqrt{1/\kappa}, \mathbf{0})^\top$ as a transition:

$$\mathbf{Z}_{\text{latent}} = \exp_{\mathbf{o}}\left((0, \mathbf{W}\mathbf{x})^\top\right), \quad \mathbf{Z}_{latent} \in \mathbb{H}^D_\kappa, \tag{5}$$

where $\mathbf{W} \in \mathbb{R}^{D\times d}$ is a learnable linear projection matrix, and $\exp_{\mathbf{o}} : \mathbb{T}_{\mathbf{o}}\mathbb{H}^D_\kappa \to \mathbb{H}^D_\kappa$ is the exponential map under the Lorentz model. Through this mapping, we obtain a high-dimensional hyperbolic latent variable $\mathbf{Z}_{latent}$ containing rich hierarchical semantics.

Orthogonal Hyperbolic Subspace Mapping. The tangent space $\mathbb{T}_{\mathbf{o}}\mathbb{H}^D_\kappa$ of the Lorentz manifold $\mathbb{H}^D_\kappa$ at the origin $\mathbf{o}$ is a linear subspace satisfying $\langle \mathbb{S}, \mathbf{o}\rangle_{\mathcal{L}} = 0$. This space is isometrically isomorphic to the D-dimensional Euclidean space $\mathbb{R}^D$, and the induced metric on it reduces to the standard Euclidean inner product. Therefore, orthogonality decomposition can be efficiently performed in the tangent space. We define a set of Tangent Space Orthogonal Mappings $\{\phi_m\}_{m=1}^M$, where $\phi_m : \mathbb{T}_{\mathbf{o}}\mathbb{H}^D_\kappa \to \mathbb{T}_{\mathbf{o}}\mathbb{H}^{d_{\text{sub}}}_\kappa$, and employ QR decomposition to construct Orthogonal Projection Matrices.

Let $\mathbf{H} \in \mathbb{R}^{D\times D}$ be a random Gaussian matrix, whose elements are independent and identically distributed according to $\mathcal{N}(0, 1)$. Performing QR decomposition on $\mathbf{H}$ yields an orthogonal matrix $\mathbf{Q}$, whose column vectors $\{\mathbf{q}_1, \ldots, \mathbf{q}_D\}$ constitute an orthonormal basis for the tangent space.

According to above definition, we partition the column vectors of $\mathbf{Q}$ into M non-overlapping subsets, with each subset containing $d_{\text{sub}} = \lfloor\sqrt{d}\rfloor$ basis vectors. The m-th orthogonal projection matrix $\mathbf{P}_m \in \mathbb{R}^{D\times d_{\text{sub}}}$ is formed by the corresponding subset of basis vectors. For the high-dimensional hyperbolic feature $\mathbb{S} \in \mathbf{Z}_{latent}$, its representation $\mathbb{S}_{\text{sub},m}$ in the m-th orthogonal subspace is obtained through the following logarithmic-exponential transformation:

$$\mathbf{v} = \log_{\mathbf{o}}(\mathbb{S}) = \begin{pmatrix} 0 \\ \bar{\mathbf{v}} \end{pmatrix}, \qquad \mathbb{S}_{\text{sub},m} = \exp_{\mathbf{o}_m}\left(\begin{pmatrix} 0 \\ \mathbf{P}_m^\top \bar{\mathbf{v}} \end{pmatrix}\right), \tag{6}$$

where $\log_{\mathbf{o}} : \mathbb{H}^D_\kappa \to \mathbb{T}_{\mathbf{o}}\mathbb{H}^D_\kappa$ is the logarithmic map, which maps points on the manifold back to the tangent space vector $\mathbf{v}$. By construction, $\mathbf{P}_i^\top \mathbf{P}_j = \mathbf{0}$ for $i \neq j$, ensuring that their column spaces are mutually orthogonal in the common tangent space. The corresponding geodesics induced by the exponential map are orthogonal at the origin.

Definition 3.1 (Tangent Subspace Overlap)**.** Let $\mathbb{S}_1, \mathbb{S}_2 \subset \{1, \ldots, C\}$ be two random feature subsets of size k. Let $I = \mathbb{S}_1 \cap \mathbb{S}_2$ denote the set of overlapping features, with cardinality $q = |I|$. We define the *overlap projection* of a sample $\mathbf{x}$ as $\mathbf{x}_I \in \mathbb{R}^q$.

Random feature sampling may cause the feature subsets used by different trees to overlap. To analyze the geometric effects of this overlap in hyperbolic representations, we examine the relationship between the overlap size and local distance expansion, and provide a theoretical analysis under Assumption A.1. Compared with the overlapping views induced by random sampling, we construct geometrically separated representation views for different trees using non-overlapping orthogonal bases in the tangent space.

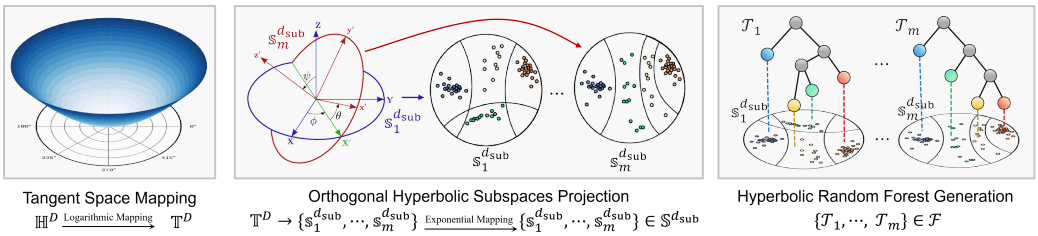


Figure 2: Random Forest Generation by Orthogonal Hyperbolic Subspaces.

Hyperbolic Random Forest Construction. After extracting the orthogonal hyperbolic subspace features $\mathbf{Z}_{\text{sub},m}$, we project them onto the Klein model. The Klein model is particularly advantageous here because its geodesics represent straight Euclidean chords, which mathematically facilitates the definition of linear hyperplanes for decision tree splitting. Subsequently, we train decision tree estimators in parallel within each subspace.

Standard decision tree algorithms employ a greedy strategy to recursively partition the feature space. Typically, the Euclidean linear midpoint of adjacent samples is calculated. However, in the Klein hyperbolic model, the space possesses anisotropic volume growth properties characterized as "center-dense and edge-sparse." The simple arithmetic mean cannot reflect the geodesic geometric center, thereby leading to severe hierarchical shifts of split boundaries at the edges of the manifold. Therefore, we introduce the Einstein Midpoint as the corrected splitting threshold.

We traverse each split node in $\mathcal{T}_m$. For the split on feature dimension k, we first identify the two sample points closest to the splitting boundary in the left and right subspaces of the node, denoted as $\mathbf{u}$ and $\mathbf{v}$ (i.e., $\mathbf{u} = \arg\max_{\mathbf{x}\in\mathcal{D}_L} x_k, \mathbf{v} = \arg\min_{\mathbf{x}\in\mathcal{D}_R} x_k$). The geometrically corrected splitting threshold on dimension k is defined using the Einstein midpoint as:

$$t_{\mathbb{H}}^{(k)} = [\text{Mid}_{\mathbb{H}}(\mathbf{u}, \mathbf{v})]_k = \frac{\gamma_{\mathbf{u}} u_k + \gamma_{\mathbf{v}} v_k}{\gamma_{\mathbf{u}} + \gamma_{\mathbf{v}}}, \tag{7}$$

where $[\cdot]_k$ denotes the k-th coordinate of a vector, u_k and v_k are the coordinates of samples $\mathbf{u}$ and $\mathbf{v}$ along this dimension, respectively, and $\gamma_{\mathbf{u}}$ and $\gamma_{\mathbf{v}}$ denote their corresponding Lorentz factors.

During the geometric construction stage, we ensemble M decision trees from different orthogonal hyperbolic subspaces to obtain the base prediction of the geometric forest:

$$F_{\text{geo}}(\mathbf{x}) = \frac{1}{M} \sum_{m=1}^{M} \mathcal{T}_m(\mathbb{S}_{\text{sub},m}), \tag{8}$$

where $F_{\text{geo}}(\mathbf{x})$ denotes the equally weighted average of the predictions from individual trees. In the complete HOT model, the graph network produces the final predictions from intermediate representations that are iteratively refined through gradient-residual-guided tree fitting, coupling tree construction with downstream task optimization.

Theorem 3.2 (Overlap-Induced Hyperbolic Expansion)**.** *For $\mathbf{x}, \mathbf{y} \in \mathcal{X}$, suppose that they are indistinguishable in the Euclidean subspace defined by the overlap I, such that $\|\mathbf{x}_I - \mathbf{y}_I\|_2 = \epsilon$ with $\epsilon \to 0$. Let $d_{\mathbb{E}}$ denote the Euclidean distance and $d_{\mathbb{K}}$ denote the Klein geodesic distance. Under Assumption A.1, the ratio of the hyperbolic distance to the Euclidean distance scales as:*

$$\lim_{\epsilon\to 0} \frac{d_{\mathbb{K}}(\phi(\mathbf{x}_I), \phi(\mathbf{y}_I))}{d_{\mathbb{E}}(\mathbf{x}_I, \mathbf{y}_I)} \propto \frac{1}{\sqrt{\delta(q)}}. \tag{9}$$

Consequently, if $\delta(q) = e^{-\lambda q}$, the expansion factor is exponential in the overlap size: $O(e^{\frac{\lambda}{2}q})$.

3.2 Rule Refinement by LLM

As illustrated in Figure 3, in order to capture the potential information and rules in the tabular data, we introduce LLMs to perform regularized extraction and refinement of decision paths, constructing a diversified set of decision tree rules. For every leaf node l in the forest, the path from the root

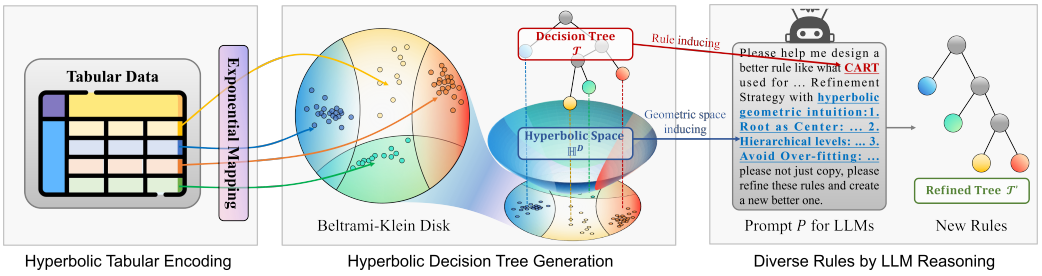


Figure 3: Illustration of how hyperbolic geometry supports decision tree construction and LLM reasoning for tabular learning. 1. Hyperbolic representations can effectively capture the latent hierarchical structure in tabular data; 2. Decision trees generated based on hyperbolic space can capture more accurate fine-grained rules; 3. The geometric prior of hyperbolic space can enhance the reasoning ability of LLMs to understand the underlying hierarchical structure in tabular data.

node to the leaf node corresponds to a logical rule. We collect the subspace coordinate splitting conditions along each root to leaf path and combine them with the splitting thresholds $t_{\mathbb{H}}^{(k)}$ adjusted by the Einstein midpoint to form the raw hyperbolic rule set $\mathcal{R}_{\text{raw}}$. Each rule describes the sample region covered by a leaf node in the corresponding subspace.

We input the raw rule set $\mathcal{R}_{\text{raw}}$ into the LLM, leveraging its semantic reasoning capabilities to perform the following optimizations. By constructing a prompt P, we guide the LLMs to understand the decision tree rule set, observe the rules from a hyperbolic perspective, and synthesize new rules from the various aggregated knowledge:

$$\mathcal{R}^* = \text{LLM}(P_{\text{meta}} \oplus P_{\mathcal{R}_{\text{raw}}} \oplus P_{\text{format}}), \tag{10}$$

where $\oplus$ denotes prompt concatenation, P_{meta} provides meta-information about the task objective, $P_{\mathcal{R}_{\text{raw}}}$ encodes the raw hyperbolic rules, and P_{format} specifies the refinement requirements and structured output format. The refined rules $\mathcal{R}^*$ serve as structural priors for subsequent gradient residual fitting by hyperbolic trees.

3.3 End-to-End Alternating Optimization via Gradient Residuals

Orthogonal hyperbolic subspaces provide geometrically separated representation views for different trees, while LLM-based rule refinement further introduces semantic priors. However, geometric diversity and semantic rules still require task-level feedback to coordinate the contributions of different trees to the final prediction. To this end, we introduce an end-to-end alternating optimization mechanism based on gradient residuals, which feeds downstream task errors back into the construction of subsequent trees, allowing geometric and semantic designs to be jointly adjusted toward the prediction objective.

Let $\mathbf{x}_i \in \mathbb{R}^d$ be the raw input feature vector. In the m-th orthogonal subspace, we denote the intermediate representation at iteration t by $\mathbf{E}_{i,m}^t \in \mathbb{H}_\kappa^{d_{\text{sub}}}$, initialized as $\mathbf{E}_{i,m}^0 = \mathbb{S}_{\text{sub},i,m}$, where $\mathbb{S}_{\text{sub},i,m}$ is the corresponding orthogonal hyperbolic subspace representation in the Lorentz model. We iteratively refine these representations through the following alternating steps, coordinating discrete semantic rules with continuous geometric updates:

Dynamic Topology Inference. Directly computing pairwise similarities among N samples in the entire dataset requires $\mathcal{O}(N^2)$ similarity comparisons. To control the graph construction cost, we dynamically construct the graph within each training mini-batch $\mathcal{B}$ at iteration t, based on the current intermediate representations $\mathbf{E}_{i,m}^t$. To prevent over-smoothing and filter out noisy connections, we employ a k-Nearest Neighbors (k-NN) approach based on hyperbolic angular similarity to construct a sparse adjacency matrix $\mathbf{A}_{\mathcal{B},m}^t$. Smooth edge weights are then assigned via a temperature-scaled exponential function to ensure stable gradient propagation. This batch-wise construction not only drastically reduces computational complexity but also acts as a powerful structural regularizer.

Graph Convolution and Gradient Backpropagation. We feed the intermediate features $\mathbf{E}_{i,m}^t$ and the dynamic graph structure $\mathbf{A}_{\mathcal{B},m}^t$ into the GNN for message passing, obtaining the final task

prediction vector $\hat{\mathbf{y}}_i$. We compute the task loss over the training mini-batch $\mathcal{B}$:

$$\mathcal{L}_{\text{task}}^t = \frac{1}{|\mathcal{B}|} \sum_{j \in \mathcal{B}} \ell(y_j, \hat{\mathbf{y}}_j), \tag{11}$$

where ℓ denotes the task-specific loss function. To guide the growth of discrete trees, we backpropagate this loss to obtain the negative gradient with respect to the intermediate representations:

$$\mathbf{g}_{i,m}^t = -\frac{\partial \mathcal{L}_{\text{task}}^t}{\partial \mathbf{E}_{i,m}^t}. \tag{12}$$

This gradient residual provides a local task-level correction signal for the current intermediate representations and serves as the pseudo-label to be fitted by subsequent hyperbolic trees. Since task prediction involves graph-based message passing, this signal incorporates both sample-specific information and neighborhood relations [25].

Semantic Rule Fitting. To incorporate gradient feedback without directly differentiating through existing tree splits, we construct a new hyperbolic decision tree $\mathcal{T}_m^{t+1}$ in the m-th subspace to fit the gradient residuals under the guidance of the refined rules $\mathcal{R}^*$:

$$\mathcal{T}_m^{t+1} = \arg\min_{\mathcal{T}} \sum_{i \in \mathcal{B}} \left\| \mathbf{g}_{i,m}^t - \psi_{\mathcal{T}}(\mathbf{E}_{i,m}^t, \mathcal{R}^*) \right\|_2^2, \tag{13}$$

where $\psi_{\mathcal{T}}$ denotes the correction response generated by tree $\mathcal{T}$ under the guidance of the refined rules. After fitting, we update the current representations using the responses of the new tree:

$$\mathbf{E}_{i,m}^{t+1} = \exp_{\mathbf{E}_{i,m}^t} \left(\eta \cdot \psi_{\mathcal{T}_m^{t+1}}(\mathbf{E}_{i,m}^t, \mathcal{R}^*) \right), \tag{14}$$

where $\eta \in (0, 1]$ is the step size that controls the magnitude of the correction. In this alternating mechanism, the GNN provides relation-aware corrective signals through the task loss, while the LLM-enhanced hyperbolic forest fits the corresponding gradient residuals under refined semantic rules, linking geometric construction and semantic refinement to the downstream prediction objective.

4 Experimental

We validated the effectiveness of the HOT framework through comprehensive evaluations on extensive real-world tabular benchmarks. We conduct analyses from multiple dimensions, including predictive performance, computational efficiency, and ablation studies, to demonstrate the model's superior performance in handling heterogeneous features and latent hierarchical structures.

4.1 Experimental Setup

4.1.1 Dataset

To comprehensively evaluate the generalization ability of the model across different task types and data scales, we selected 13 representative public tabular datasets, covering binary classification, multi-class classification, and regression tasks. Specifically, these include the 13 benchmark datasets used in the HOT framework: Blood (BL) [45], Credit (CR) [23], Car [3], Bank (BA) [34], Adult (AD) [28], Jannis (JA) [18], Cpu_act (CP) [18], Credit_reg (CRR)[18], California_housing (CA) [37], House_16H (HO) [28], Fried (FR) [5], Diamond (DI) [8], and Heart. These datasets exhibit significant diversity across different sample sizes, feature dimensions, and feature types. Following previous research [17], we use Area Under Curve(AUC) to evaluate binary and multi-class classification problems, and use Normalized Root Mean Square Error(NRMSE) to evaluate regression tasks.

4.1.2 Baselines and Details

We conducted a comparative analysis of HOT with other prominent methods in the field of tabular prediction, categorized into two groups based on whether they incorporate LLMs.

Non LLM-based method. These cover a range from classical statistical learning models to cutting-edge deep neural networks. We selected KNN [12], CART [32], MLP [30] and representative

Table 1: Test AUC (↑) performance of HOT and LLM-based baseline methods on classification tasks. The best results are highlighted in **bold**, and the second-best results are underlined.

Num	Model	BL	CR	Car	BA	AD	JA	Heart
All	TabLLM	0.700	0.723	0.991	0.921	0.920	**0.853**	0.872
	LIFT	0.675	0.680	0.880	0.816	0.805	0.730	0.835
	TP-BERT	0.710	0.715	0.940	0.912	0.835	0.781	0.822
	FeatLLM	0.715	0.779	0.942	0.930	0.885	0.802	0.895
	DeLTa	0.781	0.782	0.982	0.924	0.916	0.845	0.860
	Ours	**0.791**	**0.801**	**0.992**	**0.931**	**0.921**	0.846	**0.910**
64	TabLLM	0.680	0.612	0.971	0.690	0.842	0.732	0.771
	LIFT	0.643	0.635	0.786	0.754	0.764	0.581	0.708
	TP-BERT	0.655	0.608	0.853	0.804	0.785	0.623	0.725
	GTL	0.624	0.593	0.845	0.784	0.763	0.594	0.745
	P2T	0.458	0.58	0.799	0.686	0.711	0.523	0.681
	FeatLLM	0.711	0.604	0.868	0.774	0.838	0.581	0.693
	DeLTa	0.729	0.614	0.969	0.811	0.869	0.721	0.701
	Ours	**0.772**	**0.660**	**0.988**	**0.818**	**0.870**	**0.773**	**0.795**

Table 2: Test NRMSE (↓) performance of HOT and LLM-based baseline methods on regression tasks. The best results are highlighted in **bold**, and the second-best results are underlined.

Num	Model	CA	HO	FR	DI	CRR	CP
All	LIFT	0.629	0.902	0.518	0.328	0.983	0.379
	TP-BERT	0.399	0.708	0.215	0.163	0.795	0.138
	DeLTa	**0.351**	0.529	**0.200**	0.130	**0.780**	0.116
	Ours	0.357	**0.527**	**0.200**	**0.129**	**0.780**	**0.114**
64	LIFT	0.810	1.320	0.700	0.470	1.050	0.610
	TP-BERT	0.704	0.920	0.717	0.445	0.928	0.645
	GTL	0.834	1.189	0.724	0.483	1.190	0.663
	P2T	0.868	1.452	0.745	0.513	1.335	0.697
	DeLTa	0.701	0.919	**0.634**	0.383	0.888	**0.382**
	Ours	**0.504**	**0.875**	0.643	**0.361**	**0.870**	0.397

ensemble method, Random Forest [6]. Additionally, we compared the widely used gradient boosting decision tree algorithms, XGBoost and CatBoost, as well as FT-Transformer, TabPFN, and ModernNCA, which are based on the Transformer architecture.

Based on LLMs methods, Structured tabular are transformed into natural language text sequences. We selected baselines including TabLLM [21], TP-BERTa [43], GTL [41], P2T [35] and FeatLLM [20]. For LIFT [13] method, following the original paper's requirements, we also include LIFT as an LLM-based baseline. Additionally, LLMs can convert tabular data into decision trees. We selected DeLTa [44] as the comparison subject. This method utilizes LLMs to optimize decision tree rules rather than directly reasoning over serialized data.

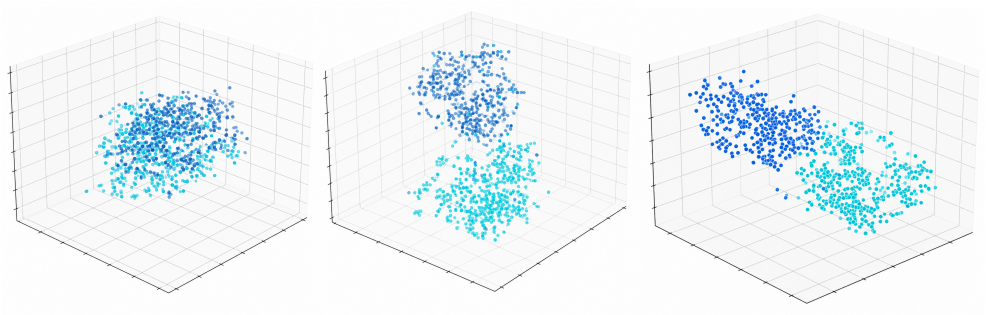
Considering that some baseline models are specifically designed for few-shot scenarios, their scalability is limited by input window length or computational resources. This makes it difficult for them to handle large-scale data tasks. Therefore, we did not include these limited models in the experimental comparisons in the main text. However, we provide a more comprehensive evaluation in the appendix. This includes performance under different training sample sizes (Num), experimental results across various large language models, and comparisons with traditional non-LLM methods. Here, Num denotes the number of training examples used, and All indicates the full training set. Results unavailable due to the aforementioned limitations are marked with a dash in the appendix tables. The experiments are run on NVIDIA A100-PCIE-40GB GPU.

4.2 Results And Analysis

We evaluated the performance comparison of HOT against mainstream LLM-based baseline methods on tabular classification and regression tasks. As shown in table 1 and table 2, in addition to standard

Table 3: Ablation Study of Different Variants. The best results are highlighted in **bold**, and the second-best results are underlined.

Variants	Blood		Credit		Car		Adult		Jannis	
	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC
w/o Hyp	0.755	0.762	0.765	0.775	0.895	0.960	0.848	0.895	0.675	0.812
w/o Ort	0.762	0.773	0.772	0.786	0.912	0.975	0.855	0.908	0.688	0.825
w/o LLM	0.769	0.785	0.780	0.795	0.925	0.988	0.861	0.915	0.696	0.838
HOT	**0.774**	**0.791**	**0.785**	**0.801**	**0.930**	**0.992**	**0.866**	**0.921**	**0.701**	**0.846**



(a) Euclidean Space. (b) Subspace_1 (c) Subspace_2

Figure 4: Visualization in Euclidean space and two orthogonal hyperbolic subspaces. Different colors represent different classes.

full-data prediction tasks, we also provided results under low-data resource conditions, simulating few-shot learning settings by randomly sampling a fixed number of training examples. The results show that HOT achieves competitive performance across classification and regression tasks under both full-data and few-shot settings, obtaining the best or second-best results on most evaluated datasets. Furthermore, we conducted an extensive comparison between HOT and non-LLM methods, covering classic tree models and deep learning models. The comparison results for regression tasks indicate that the average ranking is superior to current state-of-the-art deep tabular models. This confirms that by combining hyperbolic geometric spaces with LLM-based rule refinement, HOT can capture the intrinsic structure of data more precisely than traditional statistical models, thereby maintaining robust and excellent performance across different data scales and task types.

Additional results under different training sample sizes and comparisons with non-LLM baselines are provided in the appendix.

4.3 Ablation Study

To validate the effectiveness of individual components in our experiment (Table 3), we conducted ablation studies. (1) w/o Hyp: We removed the hyperbolic mapping and constructed the forest using orthogonal subspaces solely in Euclidean. (2) w/o Ort: We retained the hyperbolic mapping, removed the orthogonal subspace projection, replacing it with random feature sampling. (3) w/o LLM: We removed LLM-based semantic rule refinement. Experimental data shows that w/o Hyp exhibits the most significant performance drop in most tasks, confirming that hyperbolic geometry effectively captures the implicit hierarchical structure of tabular data. whereas relying solely on Euclidean space leads to a significant loss in representation capability. In contrast, the w/o Ort variant, which removes orthogonal subspace projection in favor of random sampling, retains hyperbolic properties but generally yields lower prediction accuracy than the full model, proving that the orthogonal subspace mechanism more effectively achieves feature decoupling and enhances the diversity of the ensemble model. The performance drop of w/o LLM indicates that LLM-based semantic rule refinement provides additional gains in predictive performance. To further examine ensemble diversity, we measure pairwise prediction disagreement among individual trees in HOT and Random Forest, with the detailed results and analysis provided in Appendix C.3.

Table 4: Training and inference runtime (in seconds) of HOT and other LLM-based baselines on the Adult dataset. The best results are highlighted in **bold**, and the second-best results are underlined.

# Num	Stage	TabLLM	LIFT	TP-BERTa	FeatLLM	HOT
All	Train	169,974.84	149,847.46	1,347.46	1147.34	**38.15**
	Inference	164.75	89,456.68	2.74	0.17	**0.08**
64	Train	274.35	184.48	374.20	1,147.34	**27.45**
	Inference	174.64	88,471.77	7.45	0.79	**0.06**

4.4 Visualization

As illustrated in the visualization, the Euclidean space representation (4a) suffers from severe feature entanglement. Due to the polynomial volume growth of Euclidean space, massive data points are inevitably squeezed together. In contrast, mapping data into hyperbolic space yields significantly clearer boundaries. This is because the exponential spatial capacity of the hyperbolic manifold naturally accommodates the complex hierarchical structures of the tabular data, preventing representation collapse and providing ample margin for classification. Furthermore, the different orthogonal subspaces exhibit distinctly diverse topological characteristics. In Subspace 1 (4b) and Subspace 2 (4c), different classes not only demonstrate clear separation but also display entirely different distribution patterns. This is because the orthogonal projection mechanism effectively decouples complex feature dependencies, forcing each subspace to capture a mutually independent geometric perspective of the latent semantics. These distinct spatial distributions suggest that orthogonal subspace projection provides different representation views for individual trees, supporting ensemble diversity.

4.5 Computational Efficiency

To evaluate the computational efficiency of HOT, we compare its training and inference time with representative LLM-based tabular learning methods on the Adult dataset under both full-data and 64-shot settings. As shown in Table 4, HOT consistently achieves lower computational costs across both settings. This efficiency stems from the integration of geometric tree construction and LLM-guided rule refinement. Specifically, HOT performs orthogonal subspace decomposition in the Euclidean tangent space and constructs decision trees through geometric splitting in the Klein model, avoiding expensive iterative Riemannian optimization during tree construction. Meanwhile, LLM reasoning is incorporated into the rule-refinement stage rather than invoked for individual test samples, reducing repeated semantic reasoning overhead. Furthermore, batch-wise sparse graph construction controls the computational cost of residual-guided optimization. Together, these designs enable HOT to combine geometric and semantic modeling with efficient training and inference.

5 Conclusion

In this paper, we propose HOT, a novel hyperbolic random forest framework for tabular data prediction. By combining orthogonal hyperbolic subspace projection with Einstein-midpoint-corrected tree splitting, HOT captures hierarchical structures and promotes ensemble diversity. Furthermore, it integrates geometric representations and LLM-refined semantic rules with graph-based task feedback through gradient-residual-guided tree fitting, enabling end-to-end alternating optimization. Experiments demonstrate that HOT maintains the rapid inference of traditional trees while achieving state-of-the-art performance in both full-data and few-shot scenarios. Despite its effectiveness, HOT faces scalability challenges on very large datasets, and this stage of work uses LLMs without much constraint. Future work will explore Chain of Thought (CoT) to refine semantic reasoning and further improve model scalability.

Acknowledgement

Corresponding author is Xingcheng Fu. This paper is supported by the National Natural Science Foundation of China (No. 62462007), the Guangxi Natural Science Foundation (No. 2025JJA170297), the Open Research Fund of Zhejiang Key Laboratory of Intelligent Education Technology and Application (No. 2025ZNJYKF007), and the Beijing Natural Science Foundation (No. JR26019). We sincerely thank all the authors for their hard work and contributions.

References

[1] Sercan Ö Arik and Tomas Pfister. Tabnet: Attentive interpretable tabular learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 6679–6687, 2021.

[2] Samuel A Assefa, Danial Dervovic, Mahmoud Mahfouz, Robert E Tillman, Prashant Reddy, and Manuela Veloso. Generating synthetic data in finance: opportunities, challenges and pitfalls. In *Proceedings of the first ACM international conference on AI in finance*, pages 1–8, 2020.

[3] Marko Bohanec and Vladislav Rajkovic. Knowledge acquisition and explanation for multi-attribute decision making. In *8th intl workshop on expert systems and their applications*, pages 59–78. Avignon France, 1988.

[4] 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, 2022.

[5] Leo Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

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

[7] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[8] Kuan-Yu Chen, Ping-Han Chiang, Hsin-Rung Chou, Ting-Wei Chen, and Tien-Hao Chang. Trompt: Towards a better deep neural network for tabular data. *arXiv preprint arXiv:2305.18446*, 2023.

[9] Tianqi Chen. Xgboost: A scalable tree boosting system. *Cornell University*, 2016.

[10] Weize Chen, Xu Han, Yankai Lin, Hexu Zhao, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. Fully hyperbolic neural networks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5672–5686, 2022.

[11] Philippe Chlenski, Ethan Turok, Antonio Moretti, and Itsik Pe'er. Fast hyperboloid decision tree algorithms. *arXiv preprint arXiv:2310.13841*, 2023.

[12] Thomas Cover and Peter Hart. Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1):21–27, 1967.

[13] Tuan Dinh, Yuchen Zeng, Ruisu Zhang, Ziqian Lin, Michael Gira, Shashank Rajput, Jy-yong Sohn, Dimitris Papailiopoulos, and Kangwook Lee. Lift: Language-interfaced fine-tuning for non-language machine learning tasks. *Advances in Neural Information Processing Systems*, 35:11763–11784, 2022.

[14] Lars Doorenbos, Pablo Márquez-Neila, Raphael Sznitman, and Pascal Mettes. Hyperbolic random forests. *arXiv preprint arXiv:2308.13279*, 2023.

[15] Dominick L Frosch, David Grande, Derjung M Tarn, and Richard L Kravitz. A decade of controversy: balancing policy with evidence in the regulation of prescription drug advertising. *American Journal of Public Health*, 100(1):24–32, 2010.

[16] Octavian Ganea, Gary Bécigneul, and Thomas Hofmann. Hyperbolic entailment cones for learning hierarchical embeddings. In *International conference on machine learning*, pages 1646–1655. PMLR, 2018.

[17] Yury Gorishniy, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. Revisiting deep learning models for tabular data. *Advances in neural information processing systems*, 34:18932–18943, 2021.

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

[19] Caglar Gulcehre, Misha Denil, Mateusz Malinowski, Ali Razavi, Razvan Pascanu, Karl Moritz Hermann, Peter Battaglia, Victor Bapst, David Raposo, Adam Santoro, et al. Hyperbolic attention networks. *arXiv preprint arXiv:1805.09786*, 2018.

[20] Sungwon Han, Jinsung Yoon, Sercan O Arik, and Tomas Pfister. Large language models can automatically engineer features for few-shot tabular learning. *arXiv preprint arXiv:2404.09491*, 2024.

[21] Stefan Hegselmann, Alejandro Buendia, Hunter Lang, Monica Agrawal, Xiaoyi Jiang, and David Sontag. Tabllm: Few-shot classification of tabular data with large language models. In *International conference on artificial intelligence and statistics*, pages 5549–5581. PMLR, 2023.

[22] Mikel Hernandez, Gorka Epelde, Ane Alberdi, Rodrigo Cilla, and Debbie Rankin. Synthetic data generation for tabular health records: A systematic review. *Neurocomputing*, 493:28–45, 2022.

[23] Hans Hofmann. Statlog (German Credit Data). UCI Machine Learning Repository, 1994. DOI: https://doi.org/10.24432/C5NC77.

[24] Xin Huang, Ashish Khetan, Milan Cvitkovic, and Zohar Karnin. Tabtransformer: Tabular data modeling using contextual embeddings. *arXiv preprint arXiv:2012.06678*, 2020.

[25] Sergei Ivanov and Liudmila Prokhorenkova. Boost then convolve: Gradient boosting meets graph neural networks. *arXiv preprint arXiv:2101.08543*, 2021.

[26] Velimir Jurdjevic. Non-euclidean elastica. *American Journal of Mathematics*, 117(1):93–124, 1995.

[27] 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. *Advances in neural information processing systems*, 30, 2017.

[28] Ron Kohavi et al. Scaling up the accuracy of naive-bayes classifiers: A decision-tree hybrid. In *Kdd*, volume 96, pages 202–207, 1996.

[29] Ludmila I Kuncheva and Christopher J Whitaker. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine learning*, 51(2):181–207, 2003.

[30] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[31] Cheng-Te Li, Yu-Che Tsai, Chih-Yao Chen, and Jay Chiehen Liao. Graph neural networks for tabular data learning: A survey with taxonomy and directions. *ACM Computing Surveys*, 58(1):1–51, 2025.

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

[33] Hariharan Manikandan, Yiding Jiang, and J Zico Kolter. Language models are weak learners. *Advances in Neural Information Processing Systems*, 36:50907–50931, 2023.

[34] Sérgio Moro, Paulo Cortez, and Paulo Rita. A data-driven approach to predict the success of bank telemarketing. *Decision Support Systems*, 62:22–31, 2014.

[35] Jaehyun Nam, Woomin Song, Seong Hyeon Park, Jihoon Tack, Sukmin Yun, Jaehyung Kim, Kyu Hwan Oh, and Jinwoo Shin. Tabular transfer learning via prompting llms. *arXiv preprint arXiv:2408.11063*, 2024.

[36] Maximillian Nickel and Douwe Kiela. Poincaré embeddings for learning hierarchical representations. *Advances in neural information processing systems*, 30, 2017.

[37] R Kelley Pace and Ronald Barry. Sparse spatial autoregressions. *Statistics & Probability Letters*, 33(3): 291–297, 1997.

[38] Steven J Rigatti. Random forest. *Journal of insurance medicine*, 47(1):31–39, 2017.

[39] Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, et al. Multitask prompted training enables zero-shot task generalization. *arXiv preprint arXiv:2110.08207*, 2021.

[40] Alexandru Tifrea, Gary Bécigneul, and Octavian-Eugen Ganea. Poincar\'e glove: Hyperbolic word embeddings. *arXiv preprint arXiv:1810.06546*, 2018.

[41] Xumeng Wen, Han Zhang, Shun Zheng, Wei Xu, and Jiang Bian. From supervised to generative: A novel paradigm for tabular deep learning with large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3323–3333, 2024.

[42] Jiahuan Yan, Jintai Chen, Chaowen Hu, Bo Zheng, Yaojun Hu, Jimeng Sun, and Jian Wu. Small models are llm knowledge triggers on medical tabular prediction. *arXiv preprint arXiv:2403.01570*, 2024.

[43] Jiahuan Yan, Bo Zheng, Hongxia Xu, Yiheng Zhu, Danny Z Chen, Jimeng Sun, Jian Wu, and Jintai Chen. Making pre-trained language models great on tabular prediction. *arXiv preprint arXiv:2403.01841*, 2024.

[44] Hangting Ye, Jinmeng Li, He Zhao, Dandan Guo, and Yi Chang. Llm meeting decision trees on tabular data. *Advances in Neural Information Processing Systems*, 38:118095–118131, 2026.

[45] I-Cheng Yeh. Blood Transfusion Service Center. UCI Machine Learning Repository, 2008. DOI: https://doi.org/10.24432/C5GS39.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We clearly state our claims regarding the HOT framework, including its geometric decoupling properties and end-to-end synergistic training mechanism, in the Abstract and Section 1 (Introduction).

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations of our work, such as the dependency on the quality of the LLM semantic prior and potential computational overheads for extremely large-scale graphs, in the Appendix.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: The theoretical derivations for the tangent space orthogonal projection and the Einstein Midpoint in the Lorentz model are provided in the Method section, with complete formal proofs included in the Appendix.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We comprehensively disclose all necessary information, including dataset preprocessing steps, specific LLMs utilized, GNN architectures, and hyperparameter grids, ensuring full reproducibility.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We provide an anonymized ZIP file containing our full source code, preprocessed datasets, and detailed README instructions in the supplementary material to faithfully reproduce all reported results.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: All critical experimental settings, including the data splitting strategy, optimizer choices, and the alternating optimization specifics, are provided in Section 4 and further detailed in the Appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We report both the mean values and the standard deviations across 5 independent runs with different random seeds for all main benchmark evaluations in our tables.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The specific hardware infrastructure used (including CPU and GPU models) and the average training time for HOT and baselines are fully documented in the Appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Our research complies entirely with the NeurIPS Code of Ethics. The tabular datasets utilized are publicly available, anonymized, and do not contain sensitive personal information.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: A discussion addressing the broader societal impacts of improving tabular data learning, including potential positive applications and dual-use risks, is included in the Appendix.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: Our work primarily introduces a geometric ensemble learning framework for tabular data and does not release models or datasets that possess a high risk for dual-use or malicious misuse.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All baseline repositories, software libraries, and public benchmark tabular datasets used in our study are properly cited, and their respective open-source licenses are respected.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: We release the anonymized source code of HOT as a new asset. The codebase is fully documented with a clear README detailing setup, environment dependencies, and execution instructions.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: Our research relies purely on publicly available tabular datasets and does not involve any form of crowdsourcing or research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: As our study does not involve research with human subjects, obtaining Institutional Review Board (IRB) approval is not applicable to this work.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: LLMs act as a core component of our HOT framework for semantic rule refinement. We explicitly declare the models used and detail the prompt engineering mechanisms within the Method section and the Appendix.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.

A Appendix

A.1 Proof

In this section, we provide the proof for the theorem presented.

Assumption A.1 (Norm-Depth Correlation)**.** Let $\phi : \mathbb{R}^q \to \mathbb{K}^q$ be the embedding function learned by the network on the overlapping subspace. We assume the norm of the embedding $r(q) = \|\phi(x_I)\|$ is strictly increasing with respect to the overlap size q (representing hierarchical depth). Specifically, we assume an asymptotic approach to the boundary:

$$r(q)^2 = 1 - \delta(q), \tag{15}$$

where $\delta(q)$ is a strictly decreasing function, typically modeled as $\delta(q) \propto e^{-\lambda q}$ for some $\lambda > 0$.

This assumption is consistent with the property of hyperbolic networks where the norm of the embedding represents the uncertainty or the depth in the tree. Root nodes (low information) are near 0, while leaf nodes (high information/overlap) are pushed towards 1.

We now restate the main theorem regarding the discriminative gain in hyperbolic space.

Theorem A.2 (Overlap-Induced Hyperbolic Expansion)**.** *Consider two samples $x, y \in \mathcal{X}$ that are "hard negatives," meaning they are indistinguishable in the Euclidean subspace defined by the overlap I, such that $\|x_I - y_I\|_2 = \epsilon$ with $\epsilon \to 0$. Let $d_{\mathbb{E}}$ denote the Euclidean distance and $d_{\mathbb{K}}$ denote the Klein geodesic distance. Under Assumption A.1, the ratio of the hyperbolic distance to the Euclidean distance scales as:*

$$\lim_{\epsilon \to 0} \frac{d_{\mathbb{K}}(\phi(x_I), \phi(y_I))}{d_{\mathbb{E}}(x_I, y_I)} \propto \frac{1}{\sqrt{\delta(q)}}. \tag{16}$$

Consequently, if $\delta(q) = e^{-\lambda q}$, the expansion factor is exponential in the overlap size: $O(e^{\frac{\lambda}{2} q})$.

Proof. Let $u = \phi(x_I)$ and $v = \phi(y_I)$ be the embeddings of the two samples in the Klein disk $\mathbb{K}^q$. Since x and y are hard negatives, we assume they lie at approximately the same hierarchical depth (same norm) but differ slightly in orientation. Let $\|u\| \approx \|v\| = r$. Let the Euclidean difference vector be $\Delta = v - u$, with $\|\Delta\|_2 = \epsilon$.

Since $\epsilon \to 0$, we can approximate the geodesic distance $d_{\mathbb{K}}(u, v)$ using the Riemannian metric tensor $g_u^{\mathbb{K}}$. The squared infinitesimal distance is given by:

$$\begin{aligned} ds^2 &= \Delta^\top g_u^{\mathbb{K}} \Delta \\ &= \Delta^\top \left(\frac{\mathbf{I}}{1 - r^2} + \frac{uu^\top}{(1 - r^2)^2} \right) \Delta \\ &= \Delta^\top \frac{\mathbf{I}}{1 - r^2} \Delta + \Delta^\top \frac{uu^\top}{(1 - r^2)^2} \Delta. \end{aligned} \tag{17}$$

For the first item, we have:

$$\Delta^\top \mathbf{I} \Delta = \|\Delta\|^2 = \epsilon^2. \tag{18}$$

For the second item, we decompose Δ into a radial component $\Delta_\parallel$ (parallel to u) and a tangential component $\Delta_\perp$ (orthogonal to u).

$$\Delta = \Delta_\parallel + \Delta_\perp. \tag{19}$$

The term $\Delta^\top (uu^\top) \Delta = (\Delta \cdot u)^2$. Since hard negatives at the same depth are primarily separated tangentially (angularly), Δ is approximately orthogonal to u, so $(\Delta \cdot u) \approx 0$.

Thus, the dominant term for tangential separation is the first term:

$$d_{\mathbb{K}}(u, v)^2 \approx \frac{\|\Delta\|^2}{1 - r^2} = \frac{\epsilon^2}{1 - r^2}. \tag{20}$$

Taking the square root:

$$d_{\mathbb{K}}(u, v) \approx \frac{\epsilon}{\sqrt{1 - r^2}}. \tag{21}$$

Now, we invoke Assumption A.1. The overlap q dictates the norm r such that $1 - r^2 = \delta(q)$. Substituting this into the equation, we have:

$$d_{\mathbb{K}}(u, v) \approx \frac{\epsilon}{\sqrt{\delta(q)}}. \tag{22}$$

Comparing this to the Euclidean distance in the embedding space (which is simply $\|\Delta\|_2 = \epsilon$):

$$\frac{d_{\mathbb{K}}(u, v)}{d_{\mathbb{E}}(u, v)} \approx \frac{1}{\sqrt{\delta(q)}}. \tag{23}$$

If we assume the standard exponential decay of uncertainty with respect to feature accumulation, $\delta(q) = Ce^{-\lambda q}$, we obtain:

$$d_{\mathbb{K}}(u, v) \approx \epsilon \cdot C^{-\frac{1}{2}} e^{\frac{\lambda}{2} q}. \tag{24}$$

This concludes the proof. □

A.2 Implications for Random Forests

Corollary A.3 (Linear Separability in Klein Space)**.** *The expansion derived in Theorem A.2 implies that as the overlap q increases, the "margin" between classes in the Klein model increases exponentially. Since geodesics in the Klein model are Euclidean straight lines (chords), a linear hyperplane $H = \{z : w^\top z + b = 0\}$ in $\mathbb{K}^d$ defines a geodesic decision boundary.*

Let $\gamma_{\mathbb{E}}$ be the margin required to separate classes in Euclidean space. If the data is not linearly separable in Euclidean space (i.e., $\epsilon < \gamma_{\mathbb{E}}$), the Random Forest split finding may fail. However, in the Klein embedding, the effective distance is scaled to $\epsilon \cdot e^{\frac{\lambda}{2} q}$. For sufficiently large overlap q, we achieve:

$$\epsilon \cdot e^{\frac{\lambda}{2} q} > \gamma_{\text{threshold}}. \tag{25}$$

This guarantees that a linear cut (which Random Forests perform efficiently) can successfully separate complex, hierarchical data structures that are inseparable in the original Euclidean space.

B Related Work

B.1 Representation Learning for Tabular Data

Although deep neural networks have achieved dominance in image and natural language processing, gradient boosting decision tree algorithms, represented by XGBoost [9] and LightGBM [27], remain the preferred methods for processing tabular data. This is due to their robustness to heterogeneous features and high training efficiency. To address the limitations of traditional tree models in end-to-end representation learning [24], numerous deep learning architectures for tabular data have emerged in recent years. For instance, TabNet [1] achieves automated feature selection by introducing an attention mechanism. Meanwhile, FT-Transformer [17] adapts the Transformer architecture to tabular data to capture feature interactions. However, most of these methods are developed in Euclidean spaces and may struggle to capture the hierarchical structures and complex topological relationships in tabular data.

B.2 LLMs and Tabular Reasoning

As LLMs demonstrate exceptional logical reasoning and few-shot generalization capabilities, a large volume of research has emerged introducing them to tabular prediction tasks. The first category consists of sequence methods. Early works such as LIFT [13] and TabLLM [21] serialized table rows into text templates and fine-tuned models like GPT-3 [7] or T0 [39]. To address cross-domain generalization and noise issues, TP-BERTa [43] and SERSAL [42] explore table tokenization and collaborative fine-tuning strategies. GTL [41] enhanced zero-shot reasoning capabilities by optimizing instructions; P2T [35] used natural language descriptions of unlabeled data to guide the model in semi-supervised learning; Summary [33] proposed a Boosting framework treating the LLM as a weak learner; FeatLLM [20] used serialized examples to generate new features to assist classification. The second category consists of tree-structure methods. This type of work returns to the logical essence

Table 5: Test AUC (↑) performance of HOT and LLM-based baseline methods on classification tasks. The best results are highlighted in **bold**, and the second-best results are underlined.

Num	Model	BL	CR	Car	BA	AD	JA	Heart
All	TabLLM	0.700	0.723	0.991	0.921	0.920	**0.853**	0.872
	LIFT	0.675	0.680	0.880	0.816	0.805	0.730	0.835
	TP-BERT	0.710	0.715	0.940	0.912	0.835	0.781	0.822
	GTL	–	–	–	–	–	–	–
	P2T	–	–	–	–	–	–	–
	FeatLLM	0.715	0.779	0.942	0.930	0.885	0.802	0.895
	DeLTa	0.781	0.782	0.982	0.924	0.916	0.845	0.86
	Ours	**0.791**	**0.801**	**0.992**	**0.931**	**0.921**	0.846	**0.910**
256	TabLLM	0.701	0.644	0.982	0.870	0.870	0.713	0.791
	LIFT	0.660	0.629	0.833	0.795	0.801	0.610	0.763
	TP-BERT	**0.774**	0.603	0.834	0.828	0.813	0.651	0.787
	GTL	–	–	–	–	–	–	–
	P2T	–	–	–	–	–	–	–
	FeatLLM	0.712	0.631	0.875	0.815	0.837	0.612	0.818
	DeLTa	0.754	0.647	0.983	0.864	0.868	0.766	0.767
	Ours	0.762	**0.652**	**0.987**	**0.871**	**0.871**	**0.771**	**0.837**
64	TabLLM	0.680	0.612	0.971	0.690	0.842	0.732	0.771
	LIFT	0.643	0.635	0.786	0.754	0.764	0.581	0.708
	TP-BERT	0.655	0.608	0.853	0.804	0.785	0.623	0.725
	GTL	0.624	0.593	0.845	0.784	0.763	0.594	0.745
	P2T	0.458	0.58	0.799	0.686	0.711	0.523	0.681
	FeatLLM	0.711	0.604	0.868	0.774	0.838	0.581	0.693
	DeLTa	0.729	0.614	0.969	0.811	0.869	0.721	0.701
	Ours	**0.772**	**0.660**	**0.988**	**0.818**	**0.870**	**0.773**	**0.795**

of tabular data, choosing to use the rule structure of decision trees. DeLTa [44] uses the LLMs as a rule optimizer to refine and correct the logical paths extracted from decision trees. Although this method avoids the length bottleneck of serialization, relying on LLMs for iterative rule generation and correction on massive tree nodes brings huge computational costs when processing large-scale datasets.

B.3 Hyperbolic Geometry and Decision Tree Algorithms

Hyperbolic geometry has been widely proven suitable for embedding tree-like or hierarchical structure data due to the characteristic that its spatial volume grows exponentially with the radius [40, 16]. This advantage has driven the rapid development of hyperbolic machine learning methods. In the field of deep learning, methods such as hyperbolic attention networks, fully hyperbolic neural networks [10], and hyperbolic graph convolutional networks [19] have emerged. In the field of decision trees, hyperbolic methods have been generalized to hyperbolic space; for instance, HoroRF proposed decision boundaries based on horospheres for classification. [14] and [11] developed algorithms that generate constant-time decision complexity using spheres or geodesics.

C Experimental

C.1 Complete Experimental Analysis

The supplementary experiments provide a more comprehensive evaluation of HOT across classification and regression tasks, different training sample sizes, and a broader range of baseline methods. Tables 5 and 7 present comparisons with LLM-based methods, while Tables 6 and 8 extend the comparisons to non-LLM methods. Overall, HOT maintains competitive predictive performance across these settings, demonstrating its applicability to heterogeneous tabular prediction tasks.

For classification, the results show that HOT performs consistently well under both full-data and limited-data conditions. This performance is consistent with the complementary roles of geometric

Table 6: Test AUC (↑) performance of HOT and non-LLM baseline methods on classification tasks. The best results are highlighted in **bold**, and the second-best results are underlined.

Methods	BL ↑	CR ↑	Car ↑	BA ↑	AD ↑	JA ↑	Heart ↑
KNN	0.7251	0.6684	0.9912	0.7721	0.8345	0.592	0.78
CART	0.7212	0.7015	0.9833	0.8998	0.851	0.5891	0.795
MLP	0.7056	0.7388	0.955	0.9125	0.8688	0.7256	0.825
Random Forest	0.7145	0.7582	0.9322	0.9056	0.8754	0.6812	0.84
XGBoost	0.702	0.741	0.991	0.9188	0.889	0.7365	0.901
CatBoost	**0.8015**	0.7765	0.9288	0.9205	0.8901	0.845	0.865
ResNet	0.6689	0.742	0.9554	0.915	0.8675	0.7389	0.815
FT-Transformer	0.7056	0.7455	0.9045	0.916	0.8755	0.8412	0.824
SAINT	0.7288	0.769	0.9765	0.918	0.872	0.7288	0.848
TabPFN (Version 2)	0.719	0.751	**0.995**	0.922	0.8585	0.7245	0.892
ModernNCA	0.7315	0.751	0.9754	0.9175	0.8865	0.748	0.851
Ours	0.791	**0.801**	0.992	**0.931**	**0.921**	**0.846**	**0.91**

Table 7: Test NRMSE (↓) performance of HOT and LLM-based baseline methods on regression tasks. The best results are highlighted in **bold**, and the second-best results are underlined.

Num	Model	CA	HO	FR	DI	CRR	CP
All	LIFT	0.629	0.902	0.518	0.328	0.983	0.379
	TP-BERT	0.399	0.708	0.215	0.163	0.795	0.138
	GTL	–	–	–	–	–	–
	P2T	–	–	–	–	–	–
	DeLTa	**0.351**	0.529	**0.200**	0.130	**0.780**	0.116
	Ours	0.357	**0.527**	**0.200**	**0.129**	**0.780**	**0.114**
256	LIFT	0.683	1.294	0.578	0.379	1.034	0.324
	TP-BERT	**0.557**	0.798	0.545	0.368	0.863	0.359
	GTL	–	–	–	–	–	–
	P2T	–	–	–	–	–	–
	DeLTa	0.568	0.799	0.397	0.277	0.841	**0.234**
	Ours	0.567	**0.743**	**0.356**	**0.240**	**0.802**	0.257
128	LIFT	0.710	1.300	0.660	0.430	1.070	0.290
	TP-BERT	**0.597**	0.880	0.654	0.390	0.903	0.431
	GTL	–	–	–	–	–	–
	P2T	–	–	–	–	–	–
	DeLTa	0.641	0.855	**0.512**	0.319	0.857	**0.247**
	Ours	0.610	**0.805**	0.536	**0.301**	**0.822**	0.261
64	LIFT	0.810	1.320	0.700	0.470	1.050	0.610
	TP-BERT	0.704	0.920	0.717	0.445	0.928	0.645
	GTL	0.834	1.189	0.724	0.483	1.190	0.663
	P2T	0.868	1.452	0.745	0.513	1.335	0.697
	DeLTa	0.701	0.919	**0.634**	0.383	0.888	**0.382**
	Ours	**0.504**	**0.875**	0.643	**0.361**	**0.870**	0.397

construction and semantic refinement. Hyperbolic representations provide a geometric inductive bias for hierarchical tree partitioning, while orthogonal subspace projection enables individual trees to explore different representation views. These distinct views help reduce redundancy among trees and support ensemble diversity. Meanwhile, LLM-based rule refinement introduces semantic priors into the decision paths, complementing the partitions learned from the available training data. Their integration provides a potential explanation for the competitive performance across different classification datasets and training budgets.

For regression, HOT also demonstrates strong overall performance across different sample sizes, suggesting that the framework can accommodate continuous prediction objectives. An important aspect of this adaptability is the gradient-residual-guided optimization mechanism. Rather than

Table 8: Test NRMSE (↓) performance of HOT and non-LLM-based baseline methods on regression tasks. The best results are highlighted in **bold**, and the second-best results are underlined.

Methods	CA ↓	HO ↓	FR ↓	DI ↓	CRR ↓	CP ↓
KNN	0.589	0.797	0.463	0.214	1.098	0.235
CART	0.710	0.843	0.646	0.350	0.820	0.253
MLP	0.463	0.595	0.207	0.307	0.824	0.132
Random Forest	0.506	0.669	0.371	0.163	0.809	0.152
XGBoost	0.404	0.617	0.229	0.138	0.785	0.129
CatBoost	0.372	0.567	0.202	0.131	**0.778**	0.121
ResNet	0.443	0.593	0.208	0.214	0.819	0.136
FT-Transformer	0.399	0.585	0.207	0.168	0.782	0.119
SAINT	0.439	0.601	0.206	0.140	0.828	0.137
TabPFN (Version 2)	**0.344**	0.553	0.200	0.128	0.796	0.138
ModernNCA	0.366	0.627	0.203	**0.128**	0.788	0.124
Ours	0.357	**0.527**	**0.200**	0.129	0.780	**0.114**

Table 9: Statistics of Classification Datasets

	BL	CR	Car	BA	AD	JA	Heart
#objects	748	1,000	1,728	4,5211	48,842	83,733	1,190
#num. features	4	7	0	7	6	54	11
#cat. features	0	13	6	9	8	0	0
#classes	2	2	4	2	2	4	2

treating refined rules as fixed predictive outputs, HOT uses them as structural priors for fitting task-level correction signals and refining continuous intermediate representations. Graph-based message passing further incorporates neighborhood information into these signals. This interaction allows geometric representations and semantic rules to remain connected to the downstream objective, helping the model adapt to different feature dependencies and target relationships.

The comparisons with non-LLM methods further demonstrate that HOT remains competitive with established tree ensembles and deep tabular models. These observations suggest that its effectiveness extends beyond the introduction of LLM-derived priors alone. Orthogonal hyperbolic subspaces provide diverse geometric representations, semantic rule refinement enriches the resulting decision structures, and downstream gradient feedback coordinates their contributions to prediction. Across different training budgets, these components offer complementary sources of information: data-driven task feedback supports adaptation to the observed labels, while geometric and semantic priors provide additional structure when labeled examples are limited. Together, the supplementary results support the effectiveness of integrating these mechanisms for heterogeneous tabular learning.

C.2 Dataset Information

We consider a variety of supervised tabular prediction tasks with heterogeneous features, including classification and regression. Specifically, the datasets used include: Blood (BL), Credit (CR), Car, Bank (BA), Adult (AD), Jannis (JA), Cpu_act (CP), Credit_reg (CRR), California_housing (CA), House_16H (HO), Fried (FR), Diamond (DI), and Heart. The properties of these datasets are summarized in Table 9 and 10. Our primary objective is to evaluate the effectiveness of our model HOT, in the context of LLM-based tabular prediction and to ensure a fair comparison with existing methods in this category. Therefore, most of our classification datasets are selected based on their widespread use in recent LLM-based tabular learning studies, such as TabLLM and FeatLLM, both of which primarily focus on classification tasks. To extend our evaluation beyond classification, we also incorporate regression datasets from the TALENT benchmark, which has recently gained attention for assessing tabular regression performance. To ensure fair comparisons, we adhere to the same preprocessing procedures as described in the TALENT benchmark for each dataset.

Table 10: Statistics of Regression Datasets

	CP	CRR	CA	HO	FR	DI
#objects	8,192	16,714	20,640	22,784	40,768	53,940
#num. features	21	10	8	16	10	6
#cat. features	0	0	0	0	0	3

Table 11: Comparison of Prediction Disagreement Between HOT and Random Forest. Higher values indicate greater ensemble diversity.

Method	BL	CR	Car	AD	JA	Heart	BA	Avg.
HOT	0.367	0.442	0.695	0.395	0.641	0.296	0.099	0.419
Random Forest	0.124	0.316	0.230	0.092	0.282	0.202	0.060	0.187

C.2.1 Baselines and Details

We conducted a comparative analysis of HOT with other prominent methods in the field of tabular prediction, categorized into two groups based on whether they incorporate LLMs.

Non LLM-based method. These cover a range from classical statistical learning models to cutting-edge deep neural networks. We selected KNN[12], CART[32], MLP[30] and representative ensemble method, Random Forest[6]. Additionally, we compared the widely used gradient boosting decision tree algorithms, XGBoost and CatBoost, as well as FT-Transformer, TabPFN, and ModernNCA, which are based on the Transformer architecture.

Based on LLMs methods, Structured tabular are transformed into natural language text sequences. We selected baselines including TabLLM[21], TP-BERTa[43], GTL[41], P2T[35] and FeatLLM[20]. For LIFT[13] method, following the original paper's requirements, we also include LIFT as an LLM-based baseline. Additionally, LLMs can convert tabular data into decision trees. We selected DeLTa[44] as the comparison subject. This method utilizes LLMs to optimize decision tree rules rather than directly reasoning over serialized data.

The experiments are run on NVIDIA A100-PCIE-40GB GPU. The experimental results are summarized in four tables: Table 5 and Table 6 present the classification performance (measured by AUC) of LLM-based and non-LLM baselines, respectively. Meanwhile, Table 7 and Table 8 report the regression performance (both measured by NRMSE) of LLM-based and non-LLM baselines, respectively.

C.2.2 Impact of Different LLMs

To investigate the influence of underlying semantic priors on model performance, we evaluate the effectiveness of several mainstream LLMs, including GPT-5.2, Gemini 2.5 Pro, Qwen3-32B, and Llama-34B, across classification and regression tasks (see Figure 5). Overall, the performance fluctuations across different LLMs are remarkably marginal, demonstrating the strong robustness and model-agnostic nature of the HOT framework.

C.3 Tree-level Ensemble Diversity

To further investigate the diversity among individual trees, we adopt the pairwise disagreement measure following Kuncheva and Whitaker [29]. This metric measures the fraction of test samples on which two trees predict different class labels, averaged over all distinct pairs of trees. Higher disagreement indicates more diverse predictive behaviors among ensemble members. Table 11 compares the tree-level diversity of HOT and Random Forest across seven classification datasets. HOT exhibits higher disagreement across all evaluated datasets, indicating that its individual trees capture more diverse predictive patterns. This can be attributed to the orthogonal subspace construction, which provides geometrically separated representation views for different trees. In contrast, conventional Random Forests rely on random feature sampling, where different trees may still utilize overlapping feature combinations and consequently learn similar partition structures. By explicitly constructing orthogonal subspaces, HOT reduces representational overlap and encourages individual trees to capture

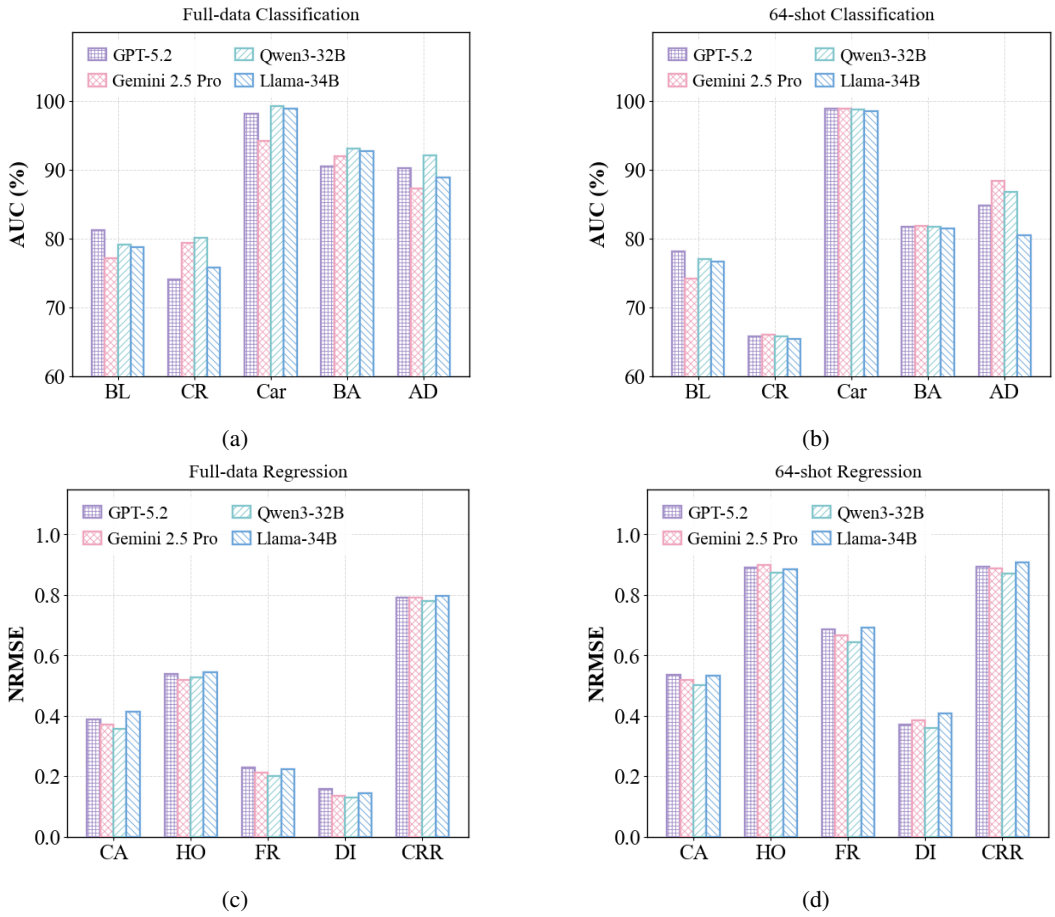


Figure 5: Performance comparison of different LLMs on classification and regression tasks under full-data and few-shot settings.

complementary feature dependencies and decision boundaries. These results provide complementary evidence for the effectiveness of orthogonal hyperbolic subspaces in promoting ensemble diversity, consistent with the performance improvements observed in the ablation study.