

HYDRA: Hyperbolic Dynamic Representation Architecture for Kolmogorov-Arnold Networks

Zhao Su¹, Yuxin Xia¹, Haoran Li², Jun Shen³, Qi Zhu⁴,
Qingguo Zhou¹, Binbin Yong¹ *

¹School of Information Science and Engineering, Lanzhou University

²Department of Data Science and AI, Monash University

³School of Computing and Information Technology, University of Wollongong

⁴College of Artificial Intelligence, Nanjing University of Aeronautics and Astronautics

Abstract

Kolmogorov-Arnold Networks (KANs) enhance nonlinear function approximation by replacing scalar weights with learnable univariate functions. However, assigning an independent function to every connection results in substantial parameter redundancy, limiting their scalability and efficiency. To reduce this redundancy, we introduce **HYperbolic Dynamic Representation Architecture (HYDRA)**, a parameter-efficient hyperbolic extension of KAN that combines spline-based functional learning with representations in the Poincaré ball. HYDRA maps vector-valued inputs into a bounded hyperbolic latent space, performs KAN-style updates in tangent space, and employs a low-rank prototype block to share functional transformations across hidden dimensions. The resulting hyperbolic representations provide a structured radial coordinate for interpretation, while radius control improves training stability by preventing boundary saturation. Extensive experiments across eight benchmark datasets demonstrate that HYDRA consistently achieves competitive or superior predictive performance while improving parameter efficiency and representation interpretability.

Introduction

Many supervised learning problems with vector-valued inputs require expressive nonlinear transformations while keeping the learned model compact and inspectable. Multilayer perceptrons (MLPs) provide flexible approximation but hide feature transformations inside dense scalar weights. Kolmogorov-Arnold Networks (KANs) replace scalar weights with learnable univariate functions on edges (Liu et al. 2025), improving local functional expressiveness but making a dense hidden-to-hidden block scale quadratically with hidden width and linearly with the number of spline bases.

The parameter-efficiency issue is closely related to how hidden representations are organized. Standard KANs operate in Euclidean spaces, where achieving stronger separation or richer representations often requires additional hidden dimensions or spline structures, increasing the parameter count. By organizing representations in a hyperbolic latent space, the model can encode variations more compactly

without substantially enlarging the KAN architecture. Meanwhile, the KAN computation is performed in the tangent space, preserving the simplicity and low-parameter nature of Euclidean spline operations. Hyperbolic representation learning provides a natural alternative because distances and volumes grow differently from Euclidean spaces (Nickel and Kiela 2017; Ganea, Bécigneul, and Hofmann 2018b). However, naive hyperbolic modeling creates a new problem: near the Poincaré-ball boundary, distances, tangent coordinates, and gradients are amplified, so a model may separate data by drifting outward rather than by learning stable functional structure.

We propose **HYperbolic Dynamic Representation Architecture (HYDRA)**, a radius-constrained low-rank neural network architecture that integrates hyperbolic representation learning with KAN-based function approximation. The key idea is to decouple representation scale from local functional modeling. Specifically, the Poincaré radius serves as a compact coordinate for encoding latent magnitude variations, while KAN splines capture local functional responses in the tangent space. HYDRA first maps normalized inputs into a bounded Poincaré ball, performs KAN-style residual updates in tangent coordinates, compresses these updates through a low-rank prototype bottleneck, and explicitly regulates the latent radius to prevent uncontrolled geometric expansion. Rather than assuming hierarchical structure in the input domain, HYDRA leverages hyperbolic geometry as an efficient and controllable representation space, while preserving the low-parameter and computationally efficient characteristics of KAN computation.

This design leads to the following contributions.

- We propose HYDRA, a hyperbolic functional learning architecture composed of multiple HYDRA blocks, which perform spline-based KAN updates in tangent space while maintaining bounded Poincaré representations.
- HYDRA introduces a low-rank prototype KAN update that reduces the dominant hidden-to-hidden parameter complexity from $O(d^2K)$ to $O(dr + r^2K)$.
- HYDRA incorporates a radius-control mechanism to constrain hyperbolic representations and mitigate unstable near-boundary effects.
- Experiments on eight datasets demonstrate that HYDRA achieves competitive or superior predictive performance

*Corresponding author: Binbin Yong (yongbb@lzu.edu.cn)

while using fewer parameters than existing approaches.

Related Work

Kolmogorov-Arnold Networks

Kolmogorov-Arnold Networks (KANs) have renewed interest in neural architectures where nonlinear transformations are represented as explicit functions rather than implicitly encoded through dense scalar weights. In the original formulation, each edge carries a learnable univariate function, typically parameterized by splines, providing a more interpretable alternative to conventional multilayer perceptrons (Liu et al. 2025). This formulation has motivated subsequent studies on KANs for scientific machine learning and broader neural architectures (Liu et al. 2024; Somvanshi et al. 2025).

Despite their interpretability advantages, dense KAN layers suffer from rapidly increasing functional parameters because each input-output connection requires an independent function. Recent variants address this limitation by modifying functional bases or introducing parameter-sharing mechanisms. Chebyshev KAN replaces spline functions with polynomial bases (Sidharth et al. 2024), Wavelet KAN introduces wavelet-based representations (Bozorgasl and Chen 2024), while FastKAN, radial basis function KAN, and parameter-reduced KAN variants further improve efficiency through alternative parameterizations (Li 2024; Ta et al. 2025). These methods reduce functional representation costs while maintaining explicit function learning. However, existing KAN variants mainly focus on edge-function parameterization, leaving the role of latent representation geometry largely unexplored. HYDRA extends this line of research by coupling KAN-based functional updates with structured hyperbolic representations.

Hyperbolic Representation Learning

Hyperbolic representation learning studies negatively curved spaces for efficient representation of complex structures. Early studies introduced Poincaré and Lorentz embeddings, demonstrating the effectiveness of hyperbolic spaces for hierarchical representation learning (Nickel and Kiela 2017, 2018). These ideas were later extended to neural computation through hyperbolic entailment regions, hyperbolic neural networks, and Riemannian optimization methods (Ganea, Bécigneul, and Hofmann 2018a,b; Bécigneul and Ganea 2019).

Recent research has expanded hyperbolic learning beyond embedding problems toward complete neural architectures, including fully hyperbolic networks, hyperbolic graph models, attention mechanisms, Transformers, supervised representation learning, and residual architectures (Shimizu, Mukuta, and Harada 2021; Chen et al. 2022; Yang et al. 2023, 2024; Nock et al. 2024; Sinha et al. 2024; Li et al. 2024; He, Yang, and Ying 2025). More recently, hyperbolic geometry has also been explored for large-model adaptation and foundation-model learning (Yang et al. 2025; He et al. 2025). However, existing hyperbolic architectures mainly focus on geometric representation learning or neural computation in curved spaces, while the interaction between hyperbolic representations and explicit functional learning remains

less explored. HYDRA explores this direction by performing KAN-based functional updates in tangent spaces associated with bounded hyperbolic representations.

Interpretable Neural Networks

Interpretability research aims to understand how neural models transform inputs into predictions. Generalized additive models and their neural extensions improve transparency by explicitly modeling feature-wise effects and interactions (Yang, Zhang, and Sudjianto 2021; Agarwal et al. 2021; Chang, Caruana, and Goldenberg 2022). Meanwhile, SHAP values provide a widely adopted model-agnostic framework for estimating feature contributions (Lundberg and Lee 2017). KANs further introduce intrinsic interpretability by representing nonlinear transformations as learnable univariate functions, enabling direct analysis of feature-response relationships.

However, functional interpretability alone may not fully characterize models with structured latent spaces. HYDRA complements functional analysis by examining hyperbolic representation dynamics to provide additional insights into model behavior.

Method

Problem Setup and Architecture

We consider supervised learning with vector-valued inputs $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$, where $\mathbf{x}_i \in \mathbb{R}^p$ is a normalized input and y_i is either a continuous target or a binary label. The goal is to learn a predictor $f_\theta(\mathbf{x})$ that is accurate, parameter-efficient, and inspectable. Dense KAN layers replace scalar weights with learnable univariate functions, but a hidden-to-hidden KAN block assigns one function to each coordinate pair. This creates a dominant $O(d^2K)$ spline cost for hidden width d and K spline bases.

HYDRA addresses this cost by combining three operations. First, it represents hidden states in a bounded Poincaré ball, so latent scale can be encoded through radius as well as direction. Second, it performs KAN-style functional learning in the tangent space, where ordinary one-dimensional spline bases remain available. Third, it compresses the tangent update through a learned prototype space before mapping the state back to the hyperbolic manifold. Figure 1 summarizes this computation.

For layer l , HYDRA maps the current hyperbolic state to the tangent space, applies a low-rank spline update, and reconstructs a bounded hyperbolic state:

$$\begin{aligned} \mathbf{z}_l &= \log_0^c(\mathbf{h}_l), \\ \mathbf{p}_l &= W_\downarrow \mathbf{z}_l, \\ \mathbf{s}_l &= \Phi_l(\mathbf{p}_l), \\ \tilde{\mathbf{z}}_{l+1} &= \mathbf{z}_l + \alpha_l W_\uparrow \mathbf{s}_l, \\ \mathbf{h}_{l+1} &= \Pi_{r_{l+1}}(\exp_0^c(\tilde{\mathbf{z}}_{l+1})), \end{aligned} \tag{1}$$

where Φ_l is a spline block in prototype coordinates, $W_\downarrow$ and $W_\uparrow$ define the bottleneck, α_l is a learned residual scale, and $\Pi_{r_{l+1}}$ enforces the layer radius budget. This formulation keeps local function learning Euclidean while making the hidden trajectory geometrically constrained and measurable.

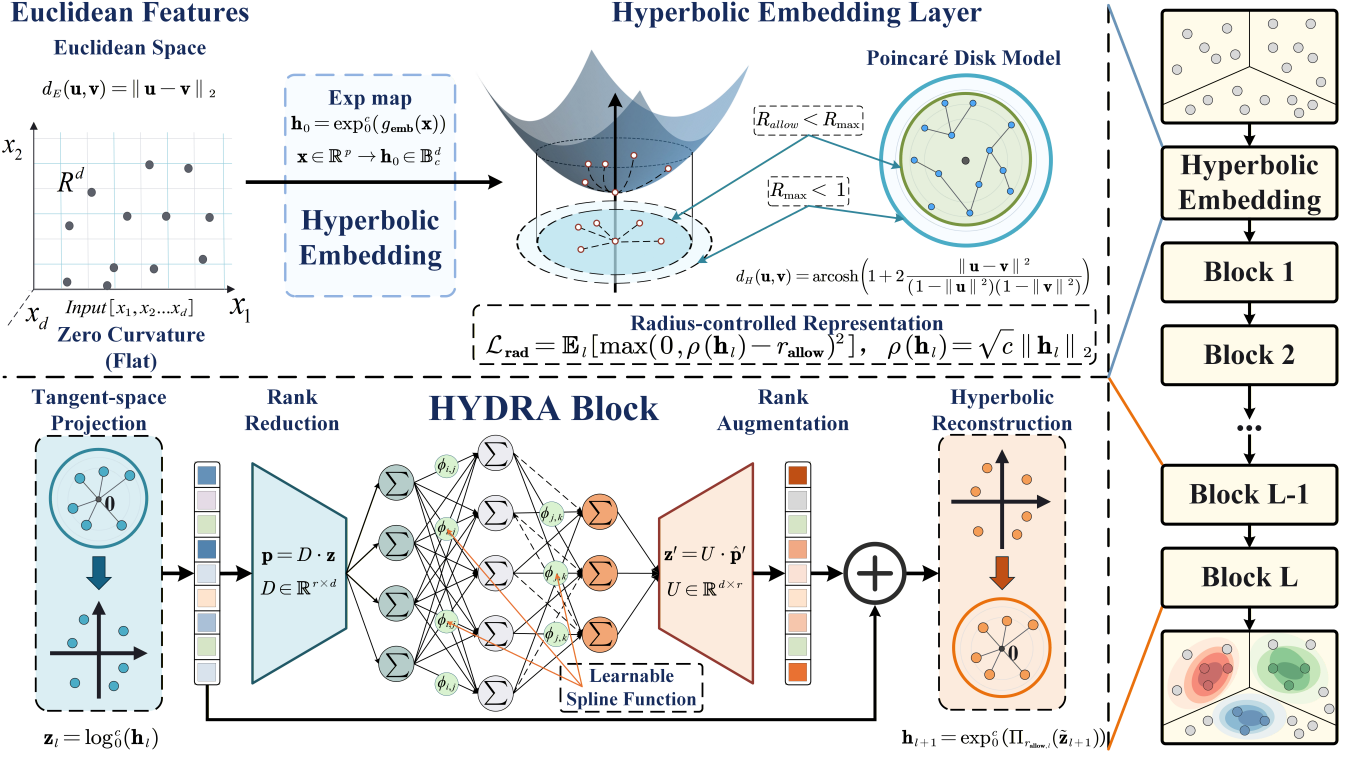


Figure 1: Overview of HYDRA. The architecture maps vector-valued features into a bounded hyperbolic representation, applies KAN-style spline updates through a low-rank prototype space, and uses radius control to stabilize geometric learning while preserving interpretable spline and radial diagnostics. All blocks share the same architecture.

Hyperbolic Embedding and Tangent-Space Updates

HYDRA represents hidden states on the Poincaré ball $\mathcal{B}_c^d = \{\mathbf{h} \in \mathbb{R}^d : c\|\mathbf{h}\|_2^2 < 1\}$ with curvature parameter $c > 0$. The normalized input is first mapped to a tangent vector $\mathbf{u}_0 = g_{\text{emb}}(\mathbf{x})$. The initial hidden state is then formed by a bounded exponential map

$$\mathbf{h}_0 = \exp_0^c(\tilde{\mathbf{u}}_0), \quad \|\mathbf{h}_0\|_c \leq r_{\text{emb}}, \quad (2)$$

where r_{emb} is the embedding radius budget. The embedding map g_{emb} is linear by default and can be replaced by a spline-based input KAN when stronger input response functions are needed.

The exponential and logarithmic maps at the origin are

$$\begin{aligned} \exp_0^c(\mathbf{v}) &= \tanh(\sqrt{c}\|\mathbf{v}\|_2) \frac{\mathbf{v}}{\sqrt{c}\|\mathbf{v}\|_2}, \\ \log_0^c(\mathbf{h}) &= \text{artanh}(\sqrt{c}\|\mathbf{h}\|_2) \frac{\mathbf{h}}{\sqrt{c}\|\mathbf{h}\|_2}. \end{aligned} \quad (3)$$

They are interpreted by continuity at the origin. These maps define a fixed coordinate chart for the KAN update. The model therefore avoids designing coordinate-wise spline functions directly on the manifold, while still returning to a hyperbolic state after each block.

Given $\mathbf{z}_t = \log_0^c(\mathbf{h}_t)$, a tangent-space KAN update has the residual form

$$\mathbf{z}_{t+1} = \mathbf{z}_t + \alpha_t \Delta_t(\mathbf{z}_t). \quad (4)$$

For a dense KAN layer, the i th coordinate is

$$\Delta_i(\mathbf{z}) = \sum_{j=1}^d \phi_{ij}(z_j), \quad \phi_{ij}(t) = \sum_{m=1}^K a_{ijm} B_m(t), \quad (5)$$

where B_m denotes spline bases and a_{ijm} denotes learned coefficients. This form is expressive and interpretable, as each edge has an explicit response curve, that leading to every input-output coordinate pair owns a separate spline. HYDRA keeps the same spline principle but moves the costly functional operator into a lower-dimensional prototype space.

Low-Rank Prototype Functional Learning

The low-rank block starts from the observation that hyperbolic radius can carry part of the latent scale variation. The tangent update need not always use all d hidden directions independently. HYDRA therefore learns an r -dimensional prototype representation, applies the KAN operator there, and lifts the result back to the hidden dimension:

$$\mathbf{p}_l = W_{\downarrow} \mathbf{z}_l, \quad \Delta_l(\mathbf{z}_l) = W_{\uparrow} \Phi_l(\mathbf{p}_l), \quad (6)$$

where

$$W_{\downarrow} \in \mathbb{R}^{r \times d}, \quad W_{\uparrow} \in \mathbb{R}^{d \times r}, \quad r \ll d. \quad (7)$$

The spline block Φ_l is applied before the up-projection, so the update is a nonlinear functional operator whose learned

Model	Regression (RMSE ↓)				Classification (Accuracy ↑)			
	CCPP	Energy	Parkinsons	Real Estate	Heart	Ionosphere	Phoneme	QSAR
MLP	3.779 (7.1k)	1.329 (1.6k)	5.304 (2.4k)	<u>6.814</u> (1.8k)	0.889 (1.4k)	0.886 (1.8k)	0.850 (1.8k)	0.882 (2.4k)
KAN	<u>3.668</u> (7.0k)	<u>0.741</u> (1.5k)	4.424 (2.4k)	8.016 (1.8k)	0.833 (1.4k)	0.843 (1.4k)	0.863 (1.8k)	0.829 (2.4k)
HGCN	4.162 (7.1k)	2.420 (1.0k)	7.733 (2.3k)	7.720 (1.8k)	0.944 (1.4k)	0.857 (1.3k)	0.831 (1.8k)	0.877 (2.3k)
HNN	4.049 (7.2k)	2.273 (1.0k)	6.643 (2.4k)	7.549 (1.8k)	0.944 (1.4k)	0.886 (1.8k)	0.832 (1.8k)	<u>0.891</u> (2.4k)
GAMI-Net	3.855 (4.8k)	1.416 (<u>1.1k</u>)	11.464 (2.4k)	7.100 (1.8k)	<u>0.926</u> (1.7k)	<u>0.943</u> (1.7k)	0.836 (1.8k)	0.858 (1.9k)
NAM	4.089 (7.1k)	3.002 (1.4k)	9.486 (2.4k)	8.891 (1.8k)	0.778 (<u>1.3k</u>)	0.800 (<u>1.1k</u>)	0.778 (1.8k)	0.796 (2.0k)
NODE-GAM	4.081 (14.4k)	1.138 (3.3k)	7.237 (8.1k)	7.251 (2.7k)	0.870 (2.6k)	0.900 (3.7k)	0.821 (6.3k)	0.853 (7.0k)
FastKAN	4.303 (<u>6.2k</u>)	2.077 (1.4k)	6.927 (2.8k)	8.394 (2.0k)	0.870 (2.3k)	0.857 (1.3k)	0.865 (1.2k)	0.839 (<u>1.6k</u>)
ChebyKAN	3.842 (9.5k)	1.023 (1.7k)	<u>4.078</u> (2.1k)	7.487 (<u>1.2k</u>)	0.889 (1.5k)	0.857 (1.2k)	<u>0.885</u> (0.9k)	0.834 (1.9k)
Wav-KAN	3.688 (8.7k)	2.047 (1.9k)	4.533 (1.8k)	9.053 (<u>1.2k</u>)	0.778 (1.4k)	0.871 (1.2k)	0.880 (<u>1.1k</u>)	0.863 (2.0k)
HYDRA (Ours)	3.604 (4.8k)	0.706 (<u>1.1k</u>)	3.534 (1.4k)	6.769 (1.0k)	0.944 (1.2k)	0.971 (0.9k)	0.885 (0.9k)	0.900 (1.5k)

Table 1: Primary metric and trainable parameters (in parentheses) on eight benchmarks. Best and second-best predictive metrics are bold and underlined; within parentheses, the smallest and second-smallest parameter counts are bold and underlined. For detailed specifications and configuration, refer to Appendix B.

spline interactions are shared through the prototype coordinates.

A full hidden-to-hidden KAN block contains approximately

$$P_{\text{full}} \approx d^2(K+1) \quad (8)$$

parameters, whereas the prototype block uses

$$P_{\text{lr}} = 2dr + r^2(K+1). \quad (9)$$

The approximate compression ratio is

$$\frac{P_{\text{lr}}}{P_{\text{full}}} \approx \frac{2}{K+1} \frac{r}{d} + \left(\frac{r}{d}\right)^2. \quad (10)$$

The low-rank prototype design reduces the functional parameter cost from $O(d^2K)$ to $O(dr + r^2K)$, where the compression ratio is mainly determined by the rank ratio r/d . Thus the dominant spline cost depends on r rather than d . A small rank is useful when the tangent update has low effective dimension, while a larger rank can be selected when the task requires more independent interactions.

Radius Control Mechanism

Hyperbolic representations can become unstable near the boundary of the Poincaré ball. In that region, distances and tangent coordinates are amplified, so an unconstrained model may reduce the loss by pushing samples outward instead of learning smoother tangent-space functions. HYDRA controls this behavior with two mechanisms. The hard projection Π_{r_l} keeps every layer within a prescribed radius. A soft penalty discourages unnecessary outward movement before the projection becomes active:

$$\mathcal{L}_{\text{rad}} = \frac{1}{L+1} \sum_{l=0}^L [\max(0, \rho(\mathbf{h}_l) - r_{\text{allow},l})]^q. \quad (11)$$

Here $\rho(\mathbf{h}_l)$ is the normalized radial coordinate, $r_{\text{allow},l}$ is the soft threshold, and q controls the penalty shape. In practice, $r_{\text{allow},l} = \tau r_l$ with $0 < \tau \leq 1$, where r_l is the hard radius budget. Projection and penalty are complementary: projection prevents boundary violations, while the penalty changes the training direction before the representation reaches the high-amplification region.

After the final HYDRA block, prediction is made from the tangent coordinate

$$\mathbf{z}_L = \log_0^c(\mathbf{h}_L), \quad \hat{y} = g_{\text{out}}(\mathbf{z}_L). \quad (12)$$

The readout g_{out} is linear by default. For regression, HYDRA minimizes mean squared error on normalized targets. For binary classification, it minimizes binary cross-entropy with logits. The complete objective is

$$\mathcal{L} = \mathcal{L}_{\text{sup}} + \lambda_{\text{rad}} \mathcal{L}_{\text{rad}} + \lambda_{\text{sp}} \mathcal{L}_{\text{sp}}, \quad (13)$$

where $\mathcal{L}_{\text{sup}}$ is the supervised loss and $\mathcal{L}_{\text{sp}}$ combines sparsity and smoothness regularization on the spline coefficients. This objective makes HYDRA a radius-constrained, low-rank functional model. Appendix A gives the universal approximation argument and additional bounds for the radius-controlled representation.

Experiments and Results

Experimental Setup

We evaluate HYDRA on eight widely used tabular benchmark datasets collected from the OpenML platform (Vanschoren et al. 2013): CCPP, Energy Heating, Parkinsons Telemonitoring, Real Estate Valuation, Heart Statlog, Ionosphere, Phoneme, and QSAR Biodegradation. These datasets cover both regression and binary classification tasks and exhibit diverse characteristics in terms of sample size, feature

dimensionality, and underlying data distributions. Such diversity provides a representative test bed for evaluating the expressive capacity, parameter efficiency, and generalization ability of HYDRA on structured tabular data. The first four datasets are formulated as regression problems, whereas the remaining four are treated as binary classification tasks.

Regression performance is evaluated using RMSE, whereas classification performance is measured using accuracy. Additional metrics are reported in Appendix B. For all experiments, the dataset is randomly split into training, validation, and test sets with ratios of 80%, 10%, and 10%, respectively, using a fixed random seed of 42.

Implementation details All experiments are conducted on a single NVIDIA L20 GPU, and all models use the same pre-processing pipeline, optimizer configuration, and evaluation protocol. Regression models predict a single scalar and are evaluated on the original target scale, while classification models output a single logit that is converted into a probability for evaluation. Beyond the task-specific loss, HYDRA introduces only two additional regularization terms: a spline regularizer that encourages smooth and compact spline functions, and a radius regularizer that constrains hidden representations within a layer-wise radius budget. Therefore, the observed performance differences mainly reflect architectural improvements rather than variations in optimization strategies, training procedures, or computational resources.

Evaluation Criteria

We select models by the primary metric and report trainable parameter count to measure compactness. HYDRA hyperparameters include hidden width, prototype rank r , spline resolution, learning rate, weight decay, radius budget, and regularization strength. The rank is treated as a compression knob: we choose the smallest rank that preserves the main metric when possible. Parameter-free geometric operations, including exponential maps, logarithmic maps, and radius projection, are not counted as trainable capacity. The parameter-efficiency claim is evaluated primarily against Euclidean KAN and MLP because they share the same dense functional or dense hidden-state modeling role; the remaining baselines are included to contextualize predictive accuracy and interpretability-oriented alternatives.

The selection rule separates predictive quality from compactness. First, we search for HYDRA configurations that are competitive on the primary metric. Second, among configurations with similar primary performance, we prefer the one with fewer trainable parameters. This avoids two misleading extremes: a large HYDRA variant that wins mainly by capacity, and a very small HYDRA variant whose parameter count is attractive but whose accuracy is no longer competitive. The reported configuration is therefore a parameter-performance trade-off rather than the largest model found during tuning. KAN and MLP are the closest Euclidean counterparts for this comparison because they represent, respectively, dense functional edges and dense hidden-state transformations.

Dataset	Task	r	Params F/L	Ratio	Primary F/L
CCPP	Reg.	3	14,536/ 1,880	0.13	3.947/ 3.919
Energy Heating	Reg.	3	1,036/ 690	0.67	2.067/ 1.959
Parkinsons TM	Reg.	3	1,211/ 485	0.40	6.342/ 6.037
Real Estate	Reg.	6	1,014/ 1,000	0.99	7.273/ 6.769
Heart Statlog	Cls.	1	12,081/ 1,091	0.09	0.889/ 0.926
Ionosphere	Cls.	7	932/ 884	0.95	0.871/ 0.943
Phoneme	Cls.	4	1,787/ 441	0.25	0.836/ 0.840
QSAR Biodeg.	Cls.	1	5,234/ 1,440	0.28	0.877/ 0.882

Table 2: Low-rank prototype ablation. F and L denote full-rank and selected low-rank HYDRA. The primary metric is RMSE for regression and Accuracy for classification.

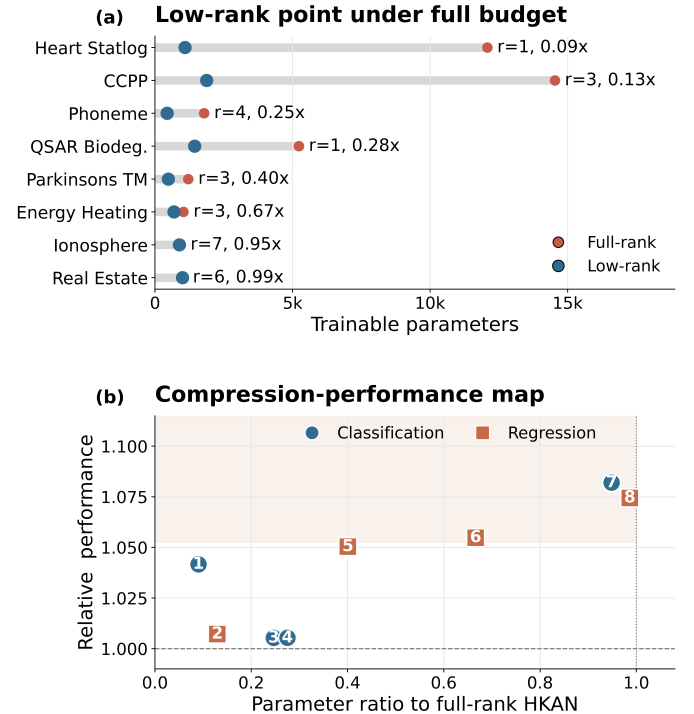


Figure 2: Low-rank ablation visualization. Selected ranks preserve the primary metric while reducing parameters relative to full-rank HYDRA. In panel (b), numbers index the datasets after sorting by increasing parameter ratio.

Main Benchmark Results

Table 1 shows that HYDRA achieves the strongest or tied-strongest primary metric on all eight datasets. On the regression tasks, HYDRA improves over KAN and MLP on CCPP, Energy Heating, Parkinsons Telemonitoring, and Real Estate Valuation while using fewer parameters than both Euclidean counterparts. On the classification tasks, HYDRA matches the best accuracy on Heart Statlog and achieves the highest accuracy on Ionosphere, Phoneme, and QSAR Biodegrada-

tion. For example, on the Parkinsons Telemonitoring dataset, HYDRA reduces RMSE from 4.424 of KAN to 3.534 while decreasing trainable parameters from 2.4k to 1.4k, corresponding to a 20.1% performance improvement with 41.7% fewer parameters. Compared with MLP, HYDRA further reduces RMSE by 33.4% under the same parameter reduction. These results support the central claim that HYDRA is not merely accurate, but accurate under a smaller trainable budget.

The benchmark mixes regression and classification because the two settings stress different aspects of the model. Regression datasets such as CCPP and Real Estate require smooth response surfaces without allocating a large number of spline edges. Classification datasets such as Ionosphere, Phoneme, and QSAR require separation while discouraging uncontrolled movement toward the Poincaré boundary. HYDRA performs well in both regimes, suggesting that the hyperbolic coordinate is not only acting as a classifier margin and not only as a regression smoother. Instead, it provides an internal scale coordinate that can be coupled with local spline response across objectives.

Parameter Efficiency

The hyperbolic maps themselves are parameter-free under fixed curvature, so HYDRA’s parameter savings must come from smaller hidden widths or lower prototype ranks. Compared with KAN and MLP, HYDRA uses fewer parameters on every dataset. This matches the low-rank analysis: if the radial coordinate absorbs part of the scale-like variation, fewer prototype directions are needed for the KAN update. The ablation below tests this interpretation against full-rank HYDRA references.

It is useful to distinguish three sources of trainable capacity. The first is the input embedding, which maps raw features into the hidden representation. The second is the hidden-to-hidden functional update. This is the dominant term for dense KAN-style models because each hidden input-output pair can own a separate spline. The third is the output read-out, which is small for the widths used here. HYDRA targets the second term: the down projection, prototype spline block, and up projection replace a dense functional map with a compact bottleneck. Thus the parameter reduction is architectural rather than a post-hoc pruning effect. When the tangent update has low effective dimension, a small rank preserves the relevant directions; when many independent interactions are needed, the selected rank grows and the savings weaken.

Ablation Studies

The low-rank ablation asks whether prototype compression preserves performance relative to a full-rank HYDRA block. The radius ablation asks whether the hyperbolic radius budget improves optimization beyond numerical safeguarding. Tables 2-3 and Figures 2-3 summarize the primary results; detailed metrics are in Appendix B.

The selected low-rank models use a mean of 46.8% and a median of 33.8% of the corresponding full-rank HYDRA parameters. As visualized in Figure 2, compression is strongest on Heart Statlog, CCPP, Phoneme, and QSAR, while Real

Dataset	Task	Primary C/U	Mean radius C/U	Δ
CCPP	Reg.	3.624 /3.690	0.450/0.646	+0.066
Energy Heating	Reg.	1.193 /1.384	0.659/0.989	+0.191
Parkinsons TM	Reg.	5.723 /5.883	0.503/0.974	+0.160
Real Estate	Reg.	6.769 /6.884	0.634/0.979	+0.115
Heart Statlog	Cls.	0.944 /0.926	0.596/0.999	+0.018
Ionosphere	Cls.	0.943 /0.871	0.679/0.923	+0.072
Phoneme	Cls.	0.856 /0.845	0.813/0.984	+0.011
QSAR Biodeg.	Cls.	0.891 /0.886	0.414/0.929	+0.005

Table 3: Radius-control ablation. C and U denote radius-controlled and unconstrained HYDRA. The primary metric is RMSE for regression and Accuracy for classification.

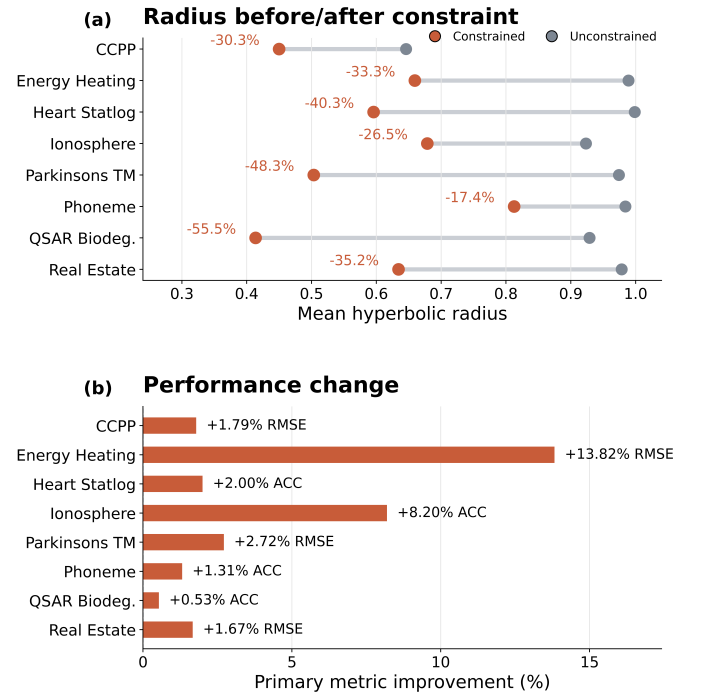


Figure 3: Radius-control ablation visualization. Constrained HYDRA reduces mean latent radius and improves the primary metric in the selected comparisons.

Estate and Ionosphere require ratios close to one, showing that the benefit is task-dependent.

Across all eight datasets, the constrained model has a smaller mean radius than the unconstrained model, and the primary metric improves in the selected comparisons. Figure 3 shows the same effect visually: radius control pulls representations inward while preserving or improving the primary metric. This matches the theoretical role of the radius budget: it bounds distance amplification and log-map gradients, discouraging near-boundary shortcuts.

Without radius control, the model may exploit the outer region of the Poincaré ball, where even small Euclidean per-

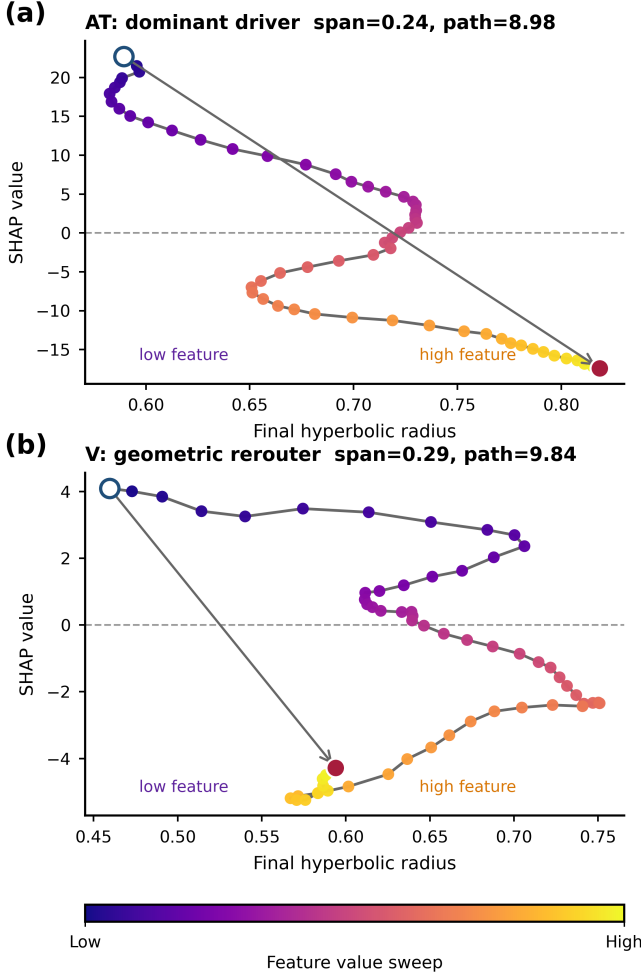


Figure 4: CCPP interpretability case study linking SHAP contribution, radius response, and path geometry.

turbations correspond to large hyperbolic distances. Such movement can help separate samples, but it can also create an unstable shortcut in which the model reduces the loss by pushing representations outward instead of learning a smooth tangent-space functional response. The constrained variant limits this behavior. The simultaneous reduction in mean radius and improvement in the primary metric suggests that radius control changes the learned representation, not only the numerical range of the hidden state.

Interpretability Analysis

HYDRA’s interpretability comes from the geometry of its hidden representation. For a controlled feature sweep, we record the final hyperbolic radius, trajectory shape, and path length of the hidden state. These quantities describe how the model reorganizes its internal representation as one input variable is changed. At the same time, we have introduced SHAP (SHapley Additive exPlanations) values as a reference to demonstrate the interpretability of the model. SHAP is a game-theoretic interpretability method that quantifies the

marginal contribution of each input feature to a model prediction. By decomposing predictions into feature-level positive or negative effects, SHAP provides an intuitive explanation of complex model behavior.

For a sweep $t \mapsto (\mathbf{x}_{-j}, t)$ with final representation $\gamma_j(t)$, the path length

$$\mathcal{L}_j = \int \|\gamma'_j(t)\|_{\mathcal{B}_c} dt, \quad \|\gamma'_j(t)\|_{\mathcal{B}_c} = \lambda_{\gamma_j(t)}^c \|\gamma'_j(t)\|_2 \quad (14)$$

weights Euclidean displacement by the local conformal factor. Large radius changes or long paths therefore indicate stronger internal reorganization.

We used the CCPP dataset as a case study to examine whether HYDRA’s latent geometry provides physically meaningful interpretability. The prediction target is net hourly electrical power output. Ambient temperature (AT), ambient pressure (AP), and relative humidity (RH) mainly describe gas-turbine operating conditions, whereas exhaust vacuum (V) reflects the steam-turbine side of the combined-cycle process. As shown in Figure 4, the AT sweep produced the clearest radius-output relation. The trajectory moved from a small-radius, positive-SHAP regime at low AT to a large-radius, negative-SHAP regime at high AT, and the one-feature PDP reduced predicted power by 37.36 MW. This agrees with the physical expectation that hotter intake air lowers air density and reduces the available mass flow through the gas turbine.

The V sweep also showed a negative output direction, but with a longer and more tortuous hyperbolic path. This pattern is consistent with a coupled steam-side condition that changes the internal representation, rather than acting as a simple direct driver. The corresponding AP and RH sweeps, reported in the Appendix, induced weaker geometric responses, consistent with their secondary roles among the operating variables. Overall, these observations support hyperbolic radius and path geometry as HYDRA-specific interpretability diagnostics. The Appendix further repeats the same visualization protocol on additional datasets to assess whether the CCPP pattern generalizes beyond this case.

Conclusion

We introduced HYDRA as a compact extension of KAN that couples tangent-space spline computation with hyperbolic latent representations. Across eight tabular benchmarks, HYDRA achieved competitive or superior predictive performance while reducing trainable parameters by 34.9% relative to Euclidean KAN and by 37.1% relative to MLP on average. Ablation studies showed that the low-rank prototype block retained predictive performance, and that radius control reduced near-boundary saturation in the Poincaré ball. The CCPP interpretability case study further showed that hyperbolic radius and latent trajectories can provide HYDRA-specific diagnostic signals that are consistent with physical expectations and SHAP trends. Overall, these results indicate that hyperbolic representation geometry can support parameter-efficient KAN-style function learning while offering an inspectable latent-space view of model behavior.

References

- Agarwal, R.; Melnick, L.; Frosst, N.; Zhang, X.; Lengerich, B.; Caruana, R.; and Hinton, G. 2021. Neural Additive Models: Interpretable Machine Learning with Neural Nets. In *Advances in Neural Information Processing Systems*, volume 34.
- Bécigneul, G.; and Ganea, O.-E. 2019. Riemannian Adaptive Optimization Methods. In *International Conference on Learning Representations*.
- Bozogasl, Z.; and Chen, H. 2024. Wav-KAN: Wavelet Kolmogorov-Arnold Networks. arXiv:2405.12832.
- Chang, C.-H.; Caruana, R.; and Goldenberg, A. 2022. NODE-GAM: Neural Generalized Additive Model for Interpretable Deep Learning. In *International Conference on Learning Representations*.
- Chen, W.; Han, X.; Lin, Y.; Zhao, H.; Liu, Z.; Li, P.; Sun, M.; and Zhou, J. 2022. Fully Hyperbolic Neural Networks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, 5672–5686.
- Ganea, O.-E.; Bécigneul, G.; and Hofmann, T. 2018a. Hyperbolic Entailment Cones for Learning Hierarchical Embeddings. In *International Conference on Machine Learning*, 1646–1655.
- Ganea, O.-E.; Bécigneul, G.; and Hofmann, T. 2018b. Hyperbolic Neural Networks. In *Advances in Neural Information Processing Systems*, volume 31.
- He, N.; Madhu, H.; Bui, N.; Yang, M.; and Ying, R. 2025. Hyperbolic Deep Learning for Foundation Models: A Survey. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 6021–6031.
- He, N.; Yang, M.; and Ying, R. 2025. Lorentzian Residual Neural Networks. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 436–447.
- Li, Y.; Mao, Y.; Yang, Y.; and Zou, D. 2024. Improving Robustness of Hyperbolic Neural Networks by Lipschitz Analysis. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 1713–1724.
- Li, Z. 2024. Kolmogorov-Arnold Networks are Radial Basis Function Networks. arXiv:2405.06721.
- Liu, Z.; Ma, P.; Wang, Y.; Matusik, W.; and Tegmark, M. 2024. KAN 2.0: Kolmogorov-Arnold Networks Meet Science. arXiv:2408.10205.
- Liu, Z.; Wang, Y.; Vaidya, S.; Ruehle, F.; Halverson, J.; Soljačić, M.; Hou, T. Y.; and Tegmark, M. 2025. KAN: Kolmogorov-Arnold Networks. In *International Conference on Learning Representations*.
- Lundberg, S.; and Lee, S.-I. 2017. A Unified Approach to Interpreting Model Predictions. In *Advances in Neural Information Processing Systems*, volume 30.
- Nickel, M.; and Kiela, D. 2017. Poincaré Embeddings for Learning Hierarchical Representations. In *Advances in Neural Information Processing Systems*, volume 30.
- Nickel, M.; and Kiela, D. 2018. Learning Continuous Hierarchies in the Lorentz Model of Hyperbolic Geometry. In *International Conference on Machine Learning*, 3779–3788.
- Nock, R.; Amid, E.; Nielsen, F.; Soen, A.; and Warmuth, M. K. 2024. Hyperbolic Embeddings of Supervised Models. In *Advances in Neural Information Processing Systems*, volume 37.
- Rudin, W. 1976. *Principles of Mathematical Analysis*. McGraw-Hill, 3rd edition.
- Shimizu, R.; Mukuta, Y.; and Harada, T. 2021. Hyperbolic Neural Networks++. In *International Conference on Learning Representations*.
- Sidharth, S. S.; Keerthana, A. R.; Gokul, R.; and Anas, K. P. 2024. Chebyshev Polynomial-Based Kolmogorov-Arnold Networks: An Efficient Architecture for Nonlinear Function Approximation. arXiv:2405.07200.
- Sinha, A.; Zeng, S.; Yamada, M.; and Zhao, H. 2024. Learning Structured Representations with Hyperbolic Embeddings. In *Advances in Neural Information Processing Systems*, volume 37.
- Somvanshi, S.; Javed, S. A.; Islam, M. M.; Pandit, D.; and Das, S. 2025. A Survey on Kolmogorov-Arnold Network. *ACM Computing Surveys*, 58(2): 1–35. Article 55.
- Ta, H.-T.; Thai, D.-Q.; Tran, A.; Sidorov, G.; and Gelbukh, A. 2025. PRKAN: Parameter-Reduced Kolmogorov-Arnold Networks. arXiv:2501.07032.
- Vanschoren, J.; van Rijn, J. N.; Bischl, B.; and Torgo, L. 2013. OpenML: Networked science in machine learning. *ACM SIGKDD Explorations Newsletter*, 15(2): 49–60.
- Yang, M.; B B, R. S.; Feng, A.; Xiong, B.; Liu, J.; King, I.; and Ying, R. 2025. Hyperbolic Fine-Tuning for Large Language Models. In *Advances in Neural Information Processing Systems*, volume 38.
- Yang, M.; Verma, H.; Zhang, D. C.; Liu, J.; King, I.; and Ying, R. 2024. Hypformer: Exploring Efficient Transformer Fully in Hyperbolic Space. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 3770–3781.
- Yang, M.; Zhou, M.; Ying, R.; Chen, Y.; and King, I. 2023. Hyperbolic Representation Learning: Revisiting and Advancing. In *International Conference on Machine Learning*, 39639–39659.
- Yang, Z.; Zhang, A.; and Sudjianto, A. 2021. GAMI-Net: An Explainable Neural Network based on Generalized Additive Models with Structured Interactions. *Pattern Recognition*, 120: 108192.

Appendix A: Universal Approximation of HYDRA

Setting. Let $K \subset \mathbb{R}^p$ be compact and $f^* \in C(K, \mathbb{R}^m)$. Define

$$\|f\|_{\infty, K} := \sup_{\mathbf{x} \in K} \|f(\mathbf{x})\|_2.$$

For $0 < r_{\max} < 1$, let

$$\mathcal{B}_{c, r_{\max}}^d = \{\mathbf{h} \in \mathbb{R}^d : \sqrt{c}\|\mathbf{h}\|_2 \leq r_{\max}\}, \quad \mathcal{T}_{c, r_{\max}}^d = \log_0^c(\mathcal{B}_{c, r_{\max}}^d).$$

On these sets,

$$\exp_0^c : \mathcal{T}_{c, r_{\max}}^d \rightarrow \mathcal{B}_{c, r_{\max}}^d, \quad \log_0^c : \mathcal{B}_{c, r_{\max}}^d \rightarrow \mathcal{T}_{c, r_{\max}}^d$$

are continuous inverses.

Proposition. For every $\varepsilon > 0$, there exist finite d, L, K_s, r and HYDRA parameters θ such that

$$\|f_\theta^{\text{HYDRA}} - f^*\|_{\infty, K} < \varepsilon.$$

Equivalently,

$$\overline{\mathcal{F}_{\text{HYDRA}}}^{\|\cdot\|_{\infty, K}} = C(K, \mathbb{R}^m),$$

provided width, depth, spline resolution, and prototype rank are allowed to increase.

Proof. Let $\varepsilon > 0$. By spline approximation and Stone-Weierstrass (Rudin 1976), there exists a finite Euclidean spline/KAN network $g_\phi : K \rightarrow \mathbb{R}^m$ such that

$$\|g_\phi - f^*\|_{\infty, K} < \varepsilon/3. \quad (\text{A1})$$

Write one Euclidean KAN residual block as

$$\mathbf{z}_{l+1} = \mathbf{z}_l + A_l \Phi_l(B_l \mathbf{z}_l), \quad (\text{A2})$$

where Φ_l is a coordinate-wise spline operator. A corresponding HYDRA block is

$$\begin{aligned} \mathbf{h}_l &= \exp_0^c(\mathbf{z}_l), \\ \tilde{\mathbf{z}}_{l+1} &= \log_0^c(\mathbf{h}_l) + W_{\uparrow, l} \Phi_l(W_{\downarrow, l} \log_0^c(\mathbf{h}_l)), \\ \mathbf{h}_{l+1} &= \Pi_{r_{\max}}(\exp_0^c(\tilde{\mathbf{z}}_{l+1})). \end{aligned} \quad (\text{A3})$$

Choose $r \geq d$ and set

$$W_{\downarrow, l} = \begin{bmatrix} I_d \\ 0 \end{bmatrix}, \quad W_{\uparrow, l} = \begin{bmatrix} A_l & 0 \end{bmatrix}. \quad (\text{A4})$$

Then

$$W_{\uparrow, l} \Phi_l(W_{\downarrow, l} \mathbf{z}) = A_l \Phi_l(\mathbf{z}), \quad (\text{A5})$$

so the low-rank prototype block contains the full tangent-space KAN update whenever $r \geq d$.

Scale the Euclidean realization by $\alpha > 0$:

$$\mathbf{z}_l^{(\alpha)} = \alpha \mathbf{z}_l, \quad \alpha K \subset \mathcal{T}_{c, r_{\max}}^d. \quad (\text{A6})$$

For $\|\mathbf{z}\| \leq R_\alpha$ with $R_\alpha \rightarrow 0$,

$$\exp_0^c(\mathbf{z}) = \mathbf{z} + O(\|\mathbf{z}\|^3), \quad \log_0^c(\mathbf{h}) = \mathbf{h} + O(\|\mathbf{h}\|^3). \quad (\text{A7})$$

Hence, for each block,

$$\sup_{\mathbf{x} \in K} \|\mathcal{H}_l^{(\alpha)}(\mathbf{x}) - \mathcal{G}_l^{(\alpha)}(\mathbf{x})\|_2 \leq C_l R_\alpha^3. \quad (\text{A8})$$

For finite depth L ,

$$\sup_{\mathbf{x} \in K} \|H_\theta^{(\alpha)}(\mathbf{x}) - G_\phi^{(\alpha)}(\mathbf{x})\|_2 \leq \sum_{l=1}^L C_l R_\alpha^3 < \varepsilon/3. \quad (\text{A9})$$

Let

$$f_\theta^{\text{HYDRA}}(\mathbf{x}) = \alpha^{-1} Q H_\theta^{(\alpha)}(\mathbf{x}), \quad g_\phi(\mathbf{x}) = \alpha^{-1} Q G_\phi^{(\alpha)}(\mathbf{x}), \quad (\text{A10})$$

and choose α so that

$$\|f_\theta^{\text{HYDRA}} - g_\phi\|_{\infty, K} < \varepsilon/3. \quad (\text{A11})$$

Combining (A1) and (A11),

$$\begin{aligned} \|f_\theta^{\text{HYDRA}} - f^*\|_{\infty, K} &\leq \|f_\theta^{\text{HYDRA}} - g_\phi\|_{\infty, K} + \|g_\phi - f^*\|_{\infty, K} \\ &< 2\varepsilon/3 < \varepsilon. \end{aligned} \quad (\text{A12})$$

Thus $\mathcal{F}_{\text{HYDRA}}$ is dense in $C(K, \mathbb{R}^m)$.

Rank condition.

$$r \geq d \implies \text{rank}(W_{\uparrow} J_\Phi(W_{\downarrow} \mathbf{z}) W_{\downarrow}) \leq d, \quad (\text{A13})$$

and the full tangent update can be recovered. Fixed $r < d$ instead imposes

$$\text{rank}(J_{\text{HYDRA}}(\mathbf{z})) \leq r, \quad (\text{A14})$$

so low-rank HYDRA is a parameter-efficient subfamily of the universal class.

Appendix B: Supplementary Experimental Results

This appendix reports secondary metrics and compact ablation details for the final benchmark datasets. The main paper uses RMSE for regression and Accuracy for classification as primary metrics. Here, regression rows additionally report MAE and R^2 , where smaller MAE and larger R^2 are better. Classification rows report F1 and ROC-AUC, where larger values are better. HYDRA is the proposed model; GAMI denotes GAMI-Net; HGCN and HNN are hyperbolic neural baselines; KAN, MLP, NAM, and NODE denote the Euclidean baseline families. Bold marks the best value in a row and underlining marks the second-best value.

For ablations, r is the selected prototype rank, and Param ratio is the low-rank/full-rank trainable-parameter ratio. F/L denotes full-rank versus selected low-rank HYDRA. C/U denotes radius-controlled versus unconstrained HYDRA. Aux. lists the secondary metrics in the same order as above. Radius P95 is the 95th percentile final radius; lower values indicate less near-boundary saturation.

Task	Dataset	HYDRA	GAMI	HGCN	HNN	KAN	MLP	NAM	NODE
<i>Regression: MAE / R^2</i>									
Reg.	CCPP	2.700/0.955	2.991/0.949	3.308/0.940	3.202/0.944	<u>2.746/0.954</u>	2.874/0.951	3.236/0.942	3.223/0.943
Reg.	Energy	0.524/0.995	1.101/0.980	1.743/0.941	1.635/0.948	<u>0.533/0.995</u>	1.062/0.982	1.924/0.910	0.859/0.987
Reg.	Parkinsons	2.479/0.892	9.563/-0.140	5.954/0.481	4.997/0.617	<u>3.211/0.830</u>	3.708/0.756	7.829/0.219	5.786/0.546
Reg.	Real Estate	4.829/0.748	4.994/0.722	5.743/0.672	5.524/0.686	5.792/0.646	<u>4.875/0.744</u>	7.242/0.564	5.286/0.710
<i>Classification: F1 / ROC-AUC</i>									
Cls.	Heart	0.933/0.970	0.909/0.949	<u>0.930/0.944</u>	<u>0.930/0.954</u>	0.780/0.924	<u>0.875/0.967</u>	0.750/0.867	0.844/0.926
Cls.	Ionosphere	0.979/0.974	<u>0.959/0.962</u>	0.900/0.949	<u>0.922/0.967</u>	0.887/0.938	0.920/0.953	0.863/0.728	0.928/0.929
Cls.	Phoneme	0.807/0.943	0.701/0.911	0.702/0.903	0.717/0.902	<u>0.772/0.924</u>	0.737/0.913	0.643/0.808	0.683/0.893
Cls.	QSAR	0.857/0.931	0.797/0.905	0.822/0.923	<u>0.841/0.919</u>	0.757/0.891	<u>0.830/0.927</u>	0.707/0.846	0.783/0.922

Table 4: Secondary metrics on the final benchmark datasets.

Task	Dataset	r	Param ratio	Primary F/L	Aux. F/L	Primary C/U	Aux. C/U	Radius P95 C/U
Reg.	CCPP	3	0.13	3.947/ 3.919	3.072/ 3.051 ; 0.946/ 0.947	3.624 /3.690	2.712 /2.757; 0.955 /0.953	0.468 /0.763
Reg.	Energy	3	0.67	2.067/ 1.959	1.580/ 1.536 ; 0.957/ 0.962	1.193 /1.384	0.878 /1.032; 0.986 /0.981	0.754 /0.999
Reg.	Parkinsons	3	0.40	6.342/ 6.037	4.737/ 4.430 ; 0.651/ 0.684	5.723 /5.883	4.043 /4.284; 0.716 /0.700	0.698 /1.000
Reg.	Real Estate	6	0.99	7.273/ 6.769	4.981/ 4.829 ; 0.709/ 0.748	6.769 /6.884	4.829 /4.922; 0.748 /0.739	0.753 /1.000
Cls.	Heart	1	0.09	0.889/ 0.926	0.875/ 0.909 ; 0.964 /0.960	0.944 /0.926	0.930 /0.909; 0.970/ 0.971	0.665 /1.000
Cls.	Ionosphere	7	0.95	0.871/ 0.943	0.907/ 0.958 ; 0.957/0.957	0.943 /0.871	0.958 /0.909; 0.966 /0.931	0.799 /1.000
Cls.	Phoneme	4	0.25	0.836/ 0.840	0.684/ 0.747 ; 0.914/0.914	0.856 /0.845	0.747 /0.722; 0.917 /0.915	0.884 /1.000
Cls.	QSAR	1	0.28	0.877/ 0.882	0.824/ 0.828 ; 0.923 /0.920	0.891 /0.886	0.844 /0.836; 0.928 /0.924	0.681 /1.000

Table 5: Combined low-rank and radius-control ablations.

Appendix C: Additional Interpretability Cases

Figure 5 extends the CCPP case study to additional datasets. Each panel links the SHAP contribution of a controlled feature sweep to the final hyperbolic radius reached by HYDRA. These cases are used as output-side references: HYDRA’s own diagnostics are the radius, trajectory, and latent-path changes, while SHAP summarizes the observed prediction contribution.

Additional interpretability cases: SHAP contribution versus hyperbolic radius

Each trajectory sweeps one feature from low to high. The x-axis tracks final hyperbolic radius; the y-axis tracks the output-level SHAP contribution.

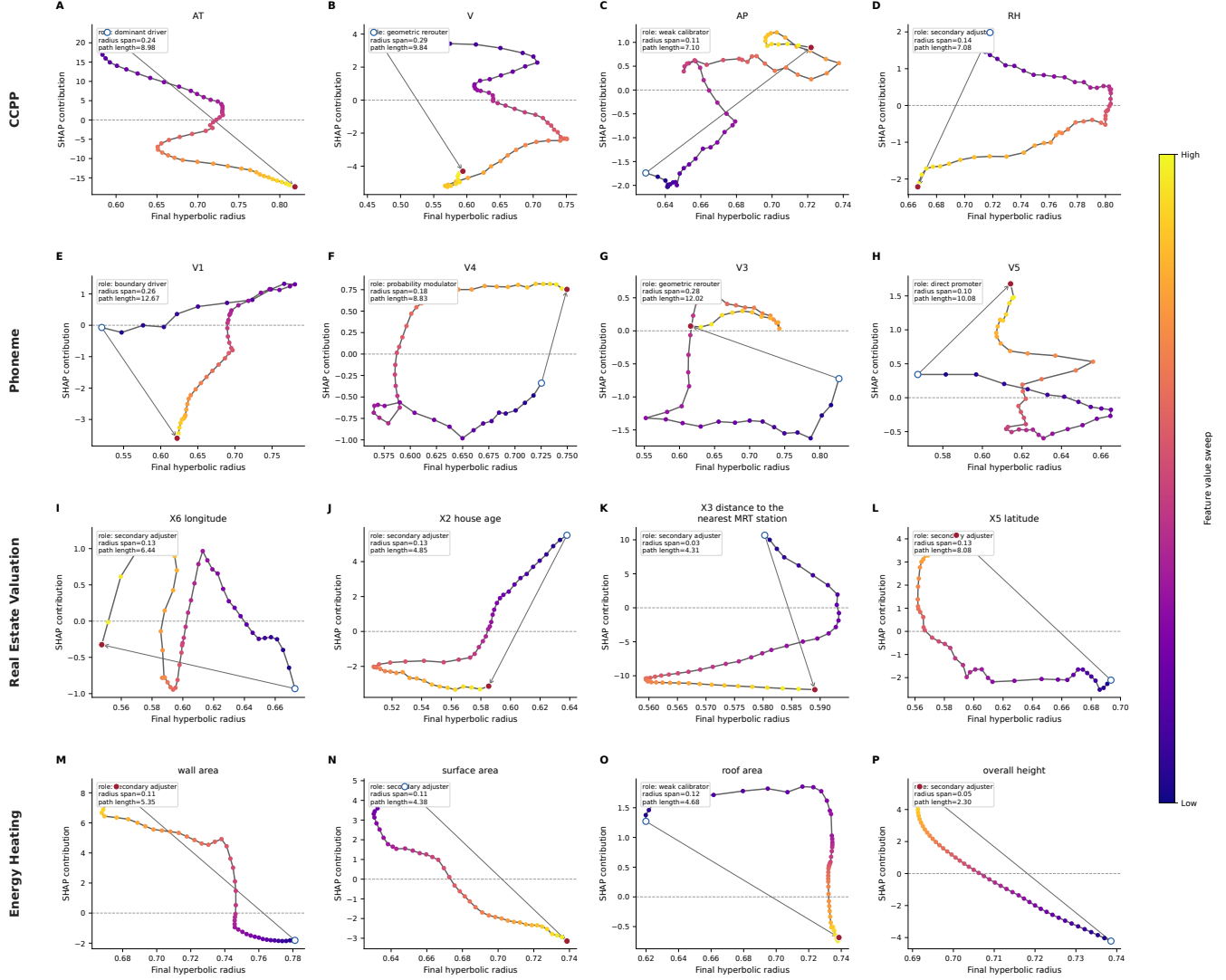


Figure 5: Additional interpretability cases beyond CCpp. Each panel traces how sweeping a representative feature changes both final hyperbolic radius and SHAP contribution.

Appendix D: Training Hyperparameters

This appendix reports the training hyperparameters used for the main comparison in Table 1. All runs use seed 42. Continuous features are normalized using training-set statistics. Regression targets are optimized in normalized space and mapped back to the original scale for reporting. Classification runs optimize binary cross-entropy with logits.

For HYDRA, Ep. is the number of training epochs and B is the mini-batch size. w is hidden width, K is the number of spline knots, and r is the low-rank prototype rank. LR and WD are the optimizer learning rate and weight decay. Drop. is dropout probability. Rad. is reported as radius-loss weight / target-radius ratio. Pos. is the positive-class loss weight; regression rows use 1.00 because no class reweighting is applied.

For compact baseline tables, each cell begins with Ep./B and then reports the architecture. KAN cells use w, K ; MLP cells use w ; HGCN and HNN cells use w, K followed by Rad. GAMI-Net cells report main subnet width, interaction subnet width, n_{int} , and main-stage/fine-tuning LR. NAM cells report shallow feature-network width, hidden feature-network width, and LR/WD; “none” means no additional hidden feature-network layer. NODE-GAM cells report trees, tree layers, tree depth, column subsampling ratio, and LR/ L_2 .

Dataset	Ep.	B	w	K	r	LR	WD	Drop.	Rad.	Pos.
CCPP	120	256	43	6	16	8.5×10^{-4}	0	0	0/0.88	1.00
Energy Heating	80	256	20	6	8	1.5×10^{-3}	10^{-4}	0	0/0.90	1.00
Parkinsons TM	220	256	13	4	12	1.8×10^{-3}	10^{-4}	0	0.03/0.88	1.00
Real Estate	140	128	28	6	6	9.0×10^{-4}	10^{-4}	0	0/0.90	1.00
Heart Statlog	80	256	47	4	2	7.0×10^{-4}	5×10^{-5}	0.02	0/0.90	1.20
Ionosphere	120	256	10	4	8	1.0×10^{-3}	10^{-4}	0	0/0.90	1.00
Phoneme	120	128	15	6	8	8.0×10^{-4}	5×10^{-5}	0.02	0.03/0.90	0.95
QSAR Biodeg.	140	256	27	6	2	6.5×10^{-4}	10^{-4}	0.015	0/0.90	1.00

Table 6: Final HYDRA hyperparameters on the eight benchmark datasets.

Dataset	KAN	MLP	HGCN	HNN
CCPP	120/256; 51,12	120/256; 56	120/256; 79,4; 0/0.90	120/256; 41,4; 0/0.90
Energy Heating	80/256; 19,12	80/256; 20	80/256; 25,4; 0/0.90	80/256; 26,4; 0/0.90
Parkinsons TM	100/512; 39,6	100/512; 28	100/512; 23,4; 0/0.90	100/512; 24,4; 0/0.90
Real Estate	80/256; 38,6	80/256; 36	80/256; 36,4; 0/0.90	80/256; 37,4; 0/0.90
Heart Statlog	80/256; 9,12	80/256; 82	80/256; 29,4; 0/0.90	80/256; 29,4; 0/0.90
Ionosphere	80/256; 13,8	80/256; 47	80/256; 21,4; 0/0.90	80/256; 27,4; 0/0.90
Phoneme	—/—; 23,8	—/—; 37	—/—; 37,4; 0/0.90	—/—; 38,4; 0/0.90
QSAR Biodeg.	—/—; 27,4	—/—; 24	—/—; 30,4; 0/0.90	—/—; 31,4; 0/0.90

Table 7: Internal neural baseline hyperparameters.

Dataset	GAMI-Net	NAM	NODE-GAM
CCPP	100/256; 12; 14–26; $10^{-3}/10^{-4}$	140/256; 21; 20–62; $5 \times 10^{-4}/0$	100/256; 160,1,6,0.5; $10^{-2}/10^{-5}$
Energy Heating	80/256; 12; 12–12; 4; $10^{-3}/10^{-4}$	140/256; 4; 32; $5 \times 10^{-4}/0$	80/256; 143,1,3,1.0; $10^{-2}/10^{-5}$
Parkinsons TM	100/256; 12; 12–12; 8; $10^{-3}/10^{-4}$	140/256; 8; 12; $5 \times 10^{-4}/10^{-6}$	100/256; 64,1,6,0.5; $10^{-2}/10^{-5}$
Real Estate	80/256; 12; 16–8; 8; $10^{-3}/10^{-4}$	116/256; 8; 16–8; $5 \times 10^{-4}/0$	80/256; 128,1,2,0.5; $10^{-2}/10^{-5}$
Heart Statlog	80/256; 32; 8–8; 4; $7 \times 10^{-4}/7 \times 10^{-5}$	140/256; 5; 15; $5 \times 10^{-4}/0$	80/256; 27,2,2,0.5; $10^{-2}/10^{-5}$
Ionosphere	80/256; 14; 7; 7; $10^{-3}/10^{-4}$	140/256; 16; none; $5 \times 10^{-4}/0$	80/256; 48,1,2,0.5; $10^{-2}/10^{-5}$
Phoneme	80/256; 12; 10–14; 8; $10^{-3}/10^{-4}$	140/256; 12; 16–8; $5 \times 10^{-4}/0$	80/256; 64,1,6,0.5; $10^{-2}/10^{-5}$
QSAR Biodeg.	80/256; 8; 8–18; 4; $10^{-3}/10^{-4}$	140/256; 24; none; $5 \times 10^{-4}/0$	80/256; 64,1,4,0.5; $10^{-2}/10^{-5}$

Table 8: External GAM baseline hyperparameters.

Appendix E: Limitations

This work has several limitations. First, our empirical evaluation focuses on standard tabular benchmarks, leaving the behavior of HYDRA on high-dimensional unstructured data (e.g., images, text, and multimodal inputs) for future investigation. Second, the proposed multi-feature analysis should be interpreted as a post hoc diagnostic of the learned model rather than a causal explanation. While it reveals that output non-additivity can coincide with hyperbolic latent reorganization, it does not establish causal or physical interactions among input variables. Finally, although we compare HYDRA with additional parameter-efficient KAN variants, its prototype rank and radius remain validation-dependent hyperparameters. Developing adaptive strategies for rank selection and radius scheduling could further improve robustness across datasets while reducing the need for manual tuning.