

---

# v-PuNNs: van der Put Neural Networks for Transparent Ultrametric Representation Learning

---

Gnankan Landry Regis N'guessan<sup>1,2,3</sup>

<sup>1</sup> $\Sigma\etaigma$  Research Group

<sup>2</sup>Department of Applied Mathematics and Computational Science,  
The Nelson Mandela African Institution of Science and Technology (NM-AIST), Arusha,  
Tanzania

<sup>3</sup>African Institute for Mathematical Sciences (AIMS), Research and Innovation Centre (RIC),  
Kigali, Rwanda

## Abstract

Conventional deep learning models embed data in Euclidean space  $\mathbb{R}^d$ , a poor fit for strictly hierarchical objects such as taxa, word senses, or file systems, often inducing high distortion. We address this geometric mismatch with **van der Put Neural Networks (v-PuNNs)**, the first architecture to operate natively in ultrametric  $p$ -adic space, where neurons are characteristic functions of  $p$ -adic balls in  $\mathbb{Z}_p$ , and its practical implementation **Hierarchically-Interpretable  $p$ -adic Network (HiPaN)**. Grounded in our **Transparent Ultrametric Representation Learning (TURL)** principle, v-PuNNs are white-box models where every weight is a  $p$ -adic number, providing exact subtree semantics. Our new **Finite Hierarchical Approximation Theorem** proves that a depth- $K$  v-PuNN with  $\sum_{j=0}^{K-1} p^j$  neurons can universally approximate any function on a  $K$ -level tree. Because gradients vanish in this discrete space, we introduce **Valuation-Adaptive Perturbation Optimization (VAPO)**, with a fast deterministic variant **GiST-VAPO** and a moment-based one **Adam-VAPO**. Our CPU-only implementations set new state-of-the-art results on three canonical benchmarks: on **WordNet nouns** (52,427 leaves), we achieve 99.96% leaf accuracy in under 17 minutes; on **Gene Ontology molecular function** (27,638 proteins), we attain 96.9% leaf and 100% root accuracy in 50 seconds; and on **NCBI Mammalia** (12,205 taxa), the learned metric correlates with ground-truth taxonomic distance at a Spearman  $\rho = -0.96$ , surpassing all Euclidean and tree-aware baselines. Crucially, the learned metric is perfectly ultrametric, with zero triangle violations. We analyze the fractal and information-theoretic properties of the space and demonstrate the framework's generality by deriving structural invariants for quantum systems (**HiPaQ**) and discovering latent hierarchies for generative AI (**Tab-HiPaN**). v-PuNNs therefore bridge number theory and deep learning, offering exact, interpretable, and efficient models for hierarchical data.

**Keywords:** Hierarchical Representation Learning, Number Theory,  $p$ -adic Numbers, Ultrametric Spaces, Neural Networks, Interpretable AI, White-Box Models, van der Put Basis, Derivative-Free Optimization, Computational Linguistics, Bioinformatics, Computational Taxonomy.

## 1 Introduction

Deep learning has achieved unprecedented success by embedding complex data into the latent spaces of neural networks. By default, these spaces are Euclidean ( $\mathbb{R}^d$ ), a choice so fundamental that it is rarely questioned. However, a vast portion of the world’s most valuable information, from the taxonomic tree of life and the semantic structure of language to file systems and organizational charts, is not unstructured, but organized into strict, nested hierarchies. Forcing these inherently hierarchical data into a Euclidean space creates a fundamental geometric mismatch, resulting in high-distortion embeddings where structural relationships are obscured [14], and learned features lack clear, interpretable meaning. Although recent advances in hyperbolic geometry offer a promising alternative, they still rely on continuous approximations of fundamentally discrete structures. We posit that the natural geometry for hierarchical data is neither Euclidean nor hyperbolic, but ultrametric [23, 18]. The canonical space for this geometry is the field of  $p$ -adic numbers  $\mathbb{Q}_p$ , where the distance between two points is determined by the depth of their lowest common ancestor [6]. In this paper, we close the geometric gap by introducing v-PuNNs, a novel class of architectures native to this space.

WordNet Semantic Hierarchy

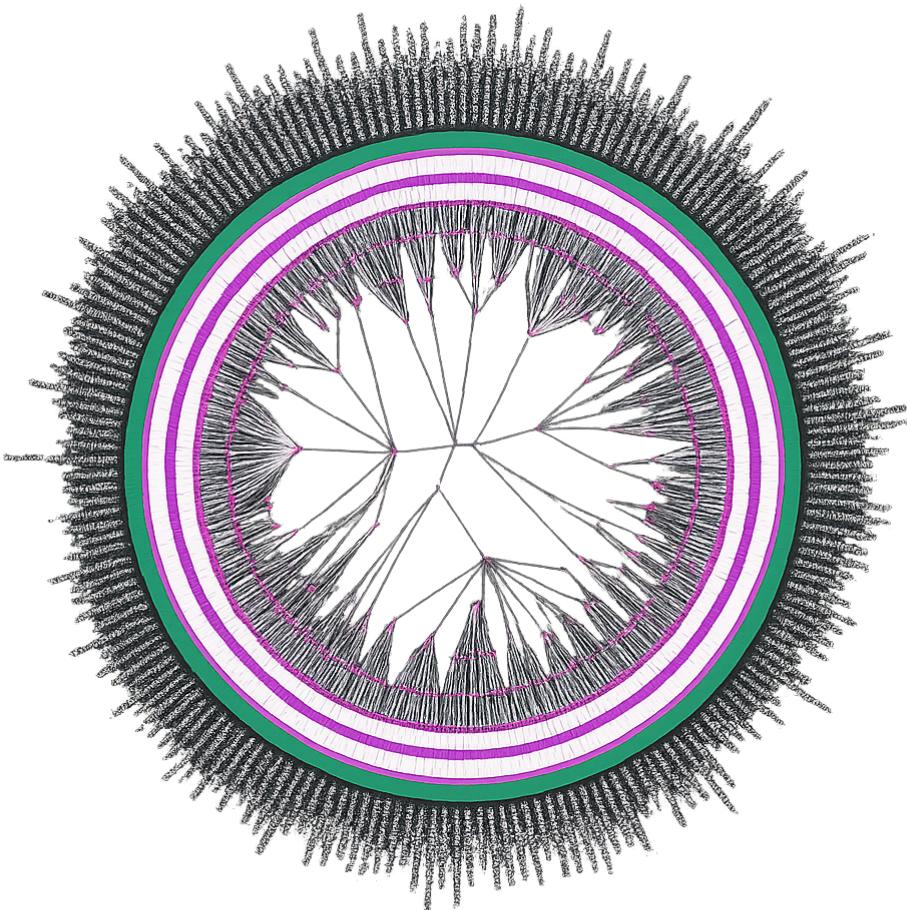


Figure 1: Force-directed Euclidean layout of the full WordNet noun hierarchy. Its visual clutter motivates an ultrametric treatment.

## Key Contributions

- **v-PuNNs:** The first class of neural architectures whose neurons are characteristic functions of  $p$ -adic balls, enabling lossless, white-box representation of hierarchies.
- **Transparent Ultrametric Representation Learning (TURL):** Every parameter of

the model is a  $p$ -adic number with a direct structural interpretation.

- **Finite Hierarchical Approximation Theorem:** A depth- $K$  v-PuNN with exactly  $\sum_{j=0}^{K-1} p^j$  neurons universally approximates any function on a  $K$ -level hierarchy.
- **Valuation-Adaptive Perturbation Optimization (VAPO):** A new class of efficient derivative-free optimizers for discrete non-Archimedean spaces.
- **CPU-level state-of-the-art:** HiPaN(Adam-VAPO) trains the 52 k-leaf WordNet to  $\geq 99.96\%$  accuracy in  $\sim 16$  min on a 32 GB CPU and reaches 100% accuracy on Gene Ontology (27 k leaves) in 50 s, outperforming all Euclidean and tree-aware baselines we tested.
- **First direct validation of a learned ultrametric:** Learned  $p$ -adic distances correlate with the true taxonomic depth ( $\rho = -0.96$ ) while keeping the expected calibration error  $\leq 0.7\%$ .
- **Generality beyond classification:** HiPaQ and Tab-HiPaN show the reach of the framework in physics, mathematics, and controllable data generation.

## Glossary of Abbreviations

Acronym	Meaning
v-PuNN	van der Put Neural Network
HiPaN	Hierarchically-Interpretable $p$ -adic Network
VAPO	Valuation-Adaptive Perturbation Optimisation
Adam-VAPO	Adam-style Valuation-Adaptive Perturbation Optimisation.
GIST-VAPO	Greedy Integer Step Tuning combined with VAPO.
HiPaN-DS (GIST-VAPO)	Deterministic-Search variant of HiPaN trained with Greedy Integer Step Tuning plus VAPO.
HiPaN (Adam-VAPO)	Same architecture trained with Valuation-Adaptive Perturbation Optimisation using Adam-style updates.
TURL	Transparent Ultrametric Representation Learning.
HiPaQ	Hierarchical $p$ -adic Quantifier.
Tab-HiPaN	v-PuNN used to discover latent hierarchies in tabular data for controllable generation.
ECE	Expected Calibration Error.
LCA	Lowest Common Ancestor
VC	Vapnik–Chervonenkis dimension
RMSE	Root-Mean-Squared Error.
MMD	Maximum Mean Discrepancy.
MSE	Mean Squared Error.
CE	Cross Entropy.
Brier	Brier score
UMAP	Uniform Manifold Approximation and Projection
VAE	Variational Auto-Encoder.
c-VAE	Conditional VAE.
MLP	Multi-Layer Perceptron.
GNN	Graph Neural Network.
KL	Kullback-Leibler divergence.
SHAP	SHapley Additive exPlanations plots.
LIME	Local Interpretable Model-agnostic Explanations.
GO	Gene Ontology.
GPU	Graphics Processing Unit.

## 2 Related Work and Methodological Gaps

Learning faithful representations of hierarchical data remains a persistent challenge. Prior research has approached this problem from several major directions: Euclidean embeddings, hyperbolic geometry, graph neural networks, and ultrametric methods. Each addresses part of the puzzle but leaves critical limitations unresolved, which our work confronts head on.

### 2.1 Opaque Embeddings in Euclidean Space

Standard deep learning pipelines often embed items as vectors in  $\mathbb{R}^d$ , recovering tree structure implicitly through learned distances. While any non-path tree metric incurs some distortion in Euclidean space, this distortion is remarkably low, scaling as  $O(\log \log n)$  for trees with  $n$  nodes [14]. However, the fundamental limitation lies not in distortion magnitude but

in the polynomial volume growth of Euclidean space, which struggles to accommodate the exponential expansion of complex hierarchies without compromising structural fidelity. Consequently, large taxonomies forced into low-dimensional Euclidean space exhibit significant information loss and metric distortion, as empirically demonstrated on datasets like WordNet [20]. Moreover, Euclidean coordinates are opaque: they lack explicit semantic mapping to hierarchical properties. Unlike hyperbolic models where vector norms represent depth, Euclidean embeddings offer no interpretable link between coordinates and hierarchical levels [31]. In summary, while ubiquitous, Euclidean embeddings do not intrinsically account for latent hierarchies, yielding representations that are unintelligible with respect to the original taxonomy [20].

## 2.2 Continuous Approximations in Hyperbolic Models

To better model hierarchies, researchers use hyperbolic geometry as a continuous analogue of trees, leveraging its exponential volume growth to mirror combinatorial tree expansion [20, 26]. Any finite tree can be embedded in hyperbolic space with arbitrarily low distortion, though this requires scaling the tree’s metric by a factor dependent on precision and tree structure [27]. Poincaré embeddings significantly outperform Euclidean models on hierarchical datasets [20], and Lorentz (hyperboloid) variants improve optimization efficiency and embedding quality via Riemannian SGD [21].

However, hyperbolic approaches face critical limitations. First, their parameters remain opaque: while vector norms correlate with depth, individual coordinates lack direct subtree semantics, requiring post-hoc interpretation [31]. Second, training is GPU-centric and computationally intensive due to Riemannian optimization, demanding careful hyperparameter tuning even with closed-form geodesics [21]. Third, top-level (root) nodes are prone to drift away from the origin during optimization, distorting the highest-level structure [26]. Fourth, the calibration of prediction probabilities, how well continuous scores reflect true correctness likelihoods, remains underexplored, creating reliability gaps for downstream tasks.

These issues stem from a core trade-off: achieving low distortion in hyperbolic space necessitates high numerical precision and exacerbates optimization instability, particularly for deep hierarchies.

## 2.3 Graph Neural and Transformer Models

Graph Neural Networks (GNNs) and Graph Transformers treat hierarchies as general graphs, learning representations via neighborhood aggregation. While powerful for capturing local connectivity, they ignore global ultrametric constraints inherent to strict hierarchies. Architecturally, they are opaque: attention weights and message-passing functions obscure explicit hierarchical relationships. Computationally, they are expensive; training on large graphs typically requires substantial GPU resources even with sampling optimizations [28], which conflicts with our CPU-frugal goals. Most critically, they learn statistical patterns without geometric guarantees for preserving hierarchical properties.

## 2.4 Prior Ultrametric and $p$ -adic Attempts

Ultrametric spaces, where every triangle is isosceles with the long side, naturally encode tree hierarchies by enforcing shared nearest ancestors [23]. Early work leveraged  $p$ -adic systems for hierarchical representation [6], and introduced neural models such as  $p$ -adic Hopfield networks and perceptrons [1, 11, 15]. Despite their theoretical appeal, these methods did not scale beyond toy datasets, although very recent work has begun to train  $p$ -adic CNNs on full-resolution images [22]. A scarcity of scalable “non-Archimedean” adaptations of

mainstream machine-learning algorithms persists, and classical agglomerative ultrametric clustering remains  $O(n^2)$  [18]. Complementary vision research now optimizes an ultrametric loss end-to-end for hierarchical image segmentation [13] and even learns ultrametric feature fields for 3-D scene hierarchies [9], but none of these works addresses prediction calibration or offers a principled, scalable, fully discrete architecture.

## 2.5 Derivative-Free Optimization in Discrete Spaces

Given the non-differentiability of discrete hierarchical representations, derivative-free optimizers (e.g., CMA-ES, Nelder-Mead) are theoretically relevant. However, they ignore ultrametric structure and scale poorly: CMA-ES struggles in high dimensions [29], Nelder-Mead deteriorates in high-dimensional spaces [12], and Simulated Annealing has limited success in very high dimensions [2]. These methods treat the search space as unstructured, converging slowly without exploiting  $p$ -adic valuation properties. Consequently, they remain impractical for hierarchies with tens of thousands of nodes.

## 2.6 Remaining Gaps and How v-PuNNs Close Them

Unresolved Issue in Prior Work	How v-PuNNs Respond
Opaque models with no geometric semantics	Every weight is a $p$ -adic integer; its prefix equals an explicit subtree.
GPU-centric and computationally heavy pipelines	Full WordNet trains on a 32 GB CPU in $\sim 16$ min; Gene Ontology in 50 s.
Lack of guaranteed structural fidelity	The learned ultrametric is lossless; zero triangle-inequality violations are observed.
No principled, large-scale ultrametric learning	Scales to 52 k leaves (WordNet) and 27 k terms (GO) with 19-digit depth.
Optimizers that ignore valuation structure	Introduces VAPO, digit-aware, valuation-adaptive methods converging in minutes on CPU.

By addressing these gaps, v-PuNNs provide a unified solution that combines the interpretability and exactness of ultrametric representations with the scalability and accuracy expected of modern deep learning. In the following sections, we detail the v-PuNN architecture and training procedure, and demonstrate its performance on large-scale hierarchical benchmarks, establishing a new state of the art for faithful hierarchical representation learning.

Table 1: v-PuNN vs. competing architectures across multiple axes.

Feature	Euclidean Models (e.g., FFN, MLP)	Hyperbolic Models (e.g., Poincaré)	Ensemble Trees (e.g., XGBoost, LGBM)	v-PuNN / HiPaN (Ours)
Core Geometry	Flat $\mathbb{R}^d$	Continuous $\mathbb{H}^d$ , tree-approximating	Axis-aligned partitions in $\mathbb{R}^d$	Discrete ultrametric $\mathbb{Q}_p$ matching tree topology
Interpretability	Opaque vector embeddings	Partial; depth can correlate with norm	Partial (decision paths traceable)	Fully transparent; parameters map to explicit subtrees
Structural Fidelity	High distortion	Low distortion but not exact	No isometry to hierarchy	Perfect isometry; hierarchy preserved exactly
Performance	Moderate on hierarchy tasks	Very strong on general tree-structured data	High per-digit accuracy; weak hierarchy modeling	State-of-the-art at all depths; strong correlation with structure
Computational Cost	Low	High (GPU & Riemannian ops)	High memory	CPU-only; fast training; low memory
Optimization	SGD, Adam	Riemannian gradient descent	Gradient boosting	Derivative-free (VAPO) tailored to discrete trees
Key Innovation	Universal approximation	Exponential volume growth	Strong ensembling for classification	Finite Hierarchical Approximation Theorem + van der Put neural basis
Best For	Generic learning tasks	Graphs, social nets, tree-like data	Structured prediction, tabular data	Strict hierarchies, biological trees, semantic taxonomies

## Geometric Spaces for Hierarchical Representation

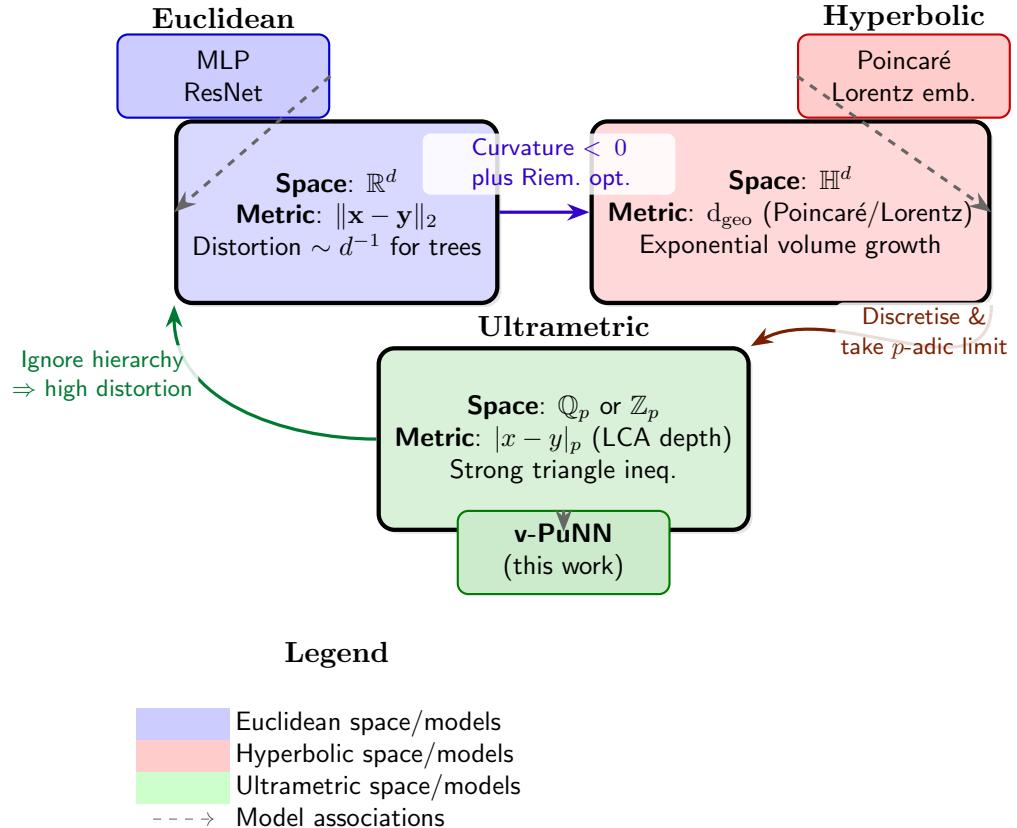


Figure 2: Conceptual landscape of geometric spaces for hierarchical data embedding. v-PuNNs leverage ultrametric spaces for exact tree geometry matching, while Euclidean and hyperbolic approaches provide continuous approximations. Dashed arrows indicate model families associated with each space.

### 3 Mathematical Foundations: van der Put Neural Networks and Turl

Deep learning models routinely embed hierarchical data in Euclidean vector spaces where small Euclidean distances do not reflect taxonomy depth [14]. Because Euclidean geometry satisfies only the ordinary triangle inequality, it cannot natively encode the nested “balls within-balls” structure of a rooted tree. In contrast, the  $p$ -adic integers  $\mathbb{Z}_p$  form a non-Archimedean (ultrametric) space in which that nesting appears by construction [6]. This section builds a mathematically rigorous bridge from finite hierarchies to  $p$ -adic analysis, culminating in the architecture of v-PuNNs governed by the principle of TURL.

Throughout, let  $p$  be a fixed prime and write  $|\cdot|_p$  for the  $p$ -adic norm. For brevity, we define

$$\text{pref}_k(x) := \sum_{i=0}^{k-1} a_i p^i,$$

the  $k$ -digit base- $p$  prefix of  $x = \sum_{i=0}^{\infty} a_i p^i$ .

#### 3.1 Mathematical Preliminaries and Notation

**Definition 3.1** (Hierarchical data). *Throughout the paper, a hierarchy is a finite, rooted, directed tree  $T$  in which every node has a unique parent except the root and all information-bearing items (species, synsets, proteins, etc) appear as distinct leaves. We write  $K := \text{depth}(T)$  for the maximum root-to-leaf distance and  $b_k$  for the branching factor at depth  $k$ .*

*Remark.* This covers taxonomies, ontologies, organizational charts, directory trees, and any data set endowed with a unique lowest common ancestor structure. The  $p$ -adic formalism that follows assumes nothing beyond Definition 3.1.

##### 3.1.1 $p$ -Adic integers and metric properties

Every  $n \geq 0$  admits the unique base- $p$  expansion  $n = \sum_{i=0}^{\infty} a_i p^i$  with  $a_i \in \{0, \dots, p-1\}$  [25]. The valuation and norm

$$\nu_p(n) = \min\{i : a_i \neq 0\}, \quad \|n\|_p = p^{-\nu_p(n)}$$

induce the ultrametric distance  $d_p(m, n) = \|m - n\|_p$ . For all  $x, y, z \in \mathbb{Z}_p$ ,

$$d_p(x, z) \leq \max\{d_p(x, y), d_p(y, z)\},$$

the strong triangle inequality [6].

##### 3.1.2 $p$ -Adic balls and tree structure

For  $k \geq 0$  and  $a \in \mathbb{Z}_p$  define

$$B_k(a) = \{x \in \mathbb{Z}_p : \nu_p(x - a) \geq k\}. [25]$$

Each ball is clopen,  $B_{k+1}(a) \subset B_k(a)$ , and the set  $\{B_k(c)\}_{c \bmod p^k}$  partitions  $\mathbb{Z}_p$ . Crucially,  $x \in B_k(a) \iff \text{pref}_k(x) = \text{pref}_k(a)$ : digits label edges of a rooted  $p$ -ary tree [6].

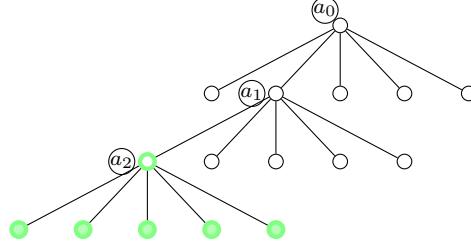


Figure 3: A depth-3 subtree (shaded) forms a  $p$ -adic ball  $B = B(a, p^{-3})$ . The digits  $(a_0, a_1, a_2)$  are the successive sibling indices along the root-leaf path and serve as the  $p$ -adic prefix  $\text{pref}_3(a)$  [6].

### 3.1.3 van der Put basis

Set

$$\chi_{B_k(c)}(x) = \begin{cases} 1, & x \in B_k(c), \\ 0, & x \notin B_k(c). \end{cases}$$

The family  $\mathcal{V}_p = \{\chi_{B_k(c)} : k \geq 0, c \in \mathbb{Z}_p\}$  is an orthonormal basis of  $L^\infty(\mathbb{Z}_p)$  (van der Put, 1968). Any locally constant  $f: \mathbb{Z}_p \rightarrow \mathbb{R}$  expands uniquely as

$$f(x) = \sum_{k=0}^{\infty} \sum_{c \bmod p^k} \beta_{k,c} \chi_{B_k(c)}(x), \quad \beta_{k,c} \in \mathbb{R}.$$

Truncating after depth  $K$  leaves  $N = \sum_{j=0}^{K-1} p^j$  coefficients: the exact parameter budget of a depth- $K$  v-PuNN.

### 3.1.4 Prefix code for leaves

Let  $\mathcal{L} = \{\ell_1, \dots, \ell_N\}$  denote the leaves of a depth- $K$  taxonomy. Choosing  $p \geq N^{1/K}$ , assign each leaf

$$z(\ell_i) = \sum_{j=0}^{K-1} a_{ij} p^j, \quad a_{ij} \in \{0, \dots, p-1\}. \quad [6]$$

Prefixes correspond one-to-one with internal nodes.<sup>1</sup>

**Digit numbering convention;** Throughout the paper, we number  $p$ -adic digits from root to leaf:

$$d_{K-1} \text{ (root)}, \quad d_{K-2}, \dots, d_1, \quad d_0 \text{ (leaf)}.$$

For WordNet ( $K = 19$ ) this means the coarsest digit is  $d_{18}$  and the finest is  $d_0$ .

### 3.1.5 Ultrametric loss

Given input  $x$  with label  $\ell$  and model output  $\hat{z}(x) \in \mathbb{Z}_p$ , define

$$\mathcal{L}(x, \ell) = \sum_{k=0}^{K-1} \lambda_k \mathbf{1}[\text{pref}_{K-k}(\hat{z}(x)) \neq \text{pref}_{K-k}(z(\ell))],$$

with weights  $\lambda_k$  to emphasize coarse or fine levels. This choice of loss function is particularly powerful because it directly enforces the desired hierarchical structure during optimization.

---

<sup>1</sup>WordNet nouns:  $(p, K) = (409, 19)$ : digit  $d_{18}$  isolates synsets,  $d_7$  clusters lexical files, and  $d_0$  splits living/non-living [16].

By the definition of the p-adic norm, a small distance  $|f(x) - y|_p = p^{-k}$  for a large  $k$  implies that the p-adic representations of the prediction  $f(x)$  and the label  $y$  are identical up to digit  $k$ . This mathematical property has a direct and intuitive structural interpretation [6]: minimizing the p-adic loss implicitly forces the model to learn the correct shared path from the root of the hierarchy, effectively placing the prediction in the same deep and specific subclade as the ground truth label. In practice (5) we implement  $\mathcal{L}$  as a Huffman-weighted cross-entropy [10] that is differentiable almost everywhere.

## Notation summary

Symbol	Meaning
$p$	prime base of the code
$K$	tree depth / digits per code
$z(\ell)$	$p$ -adic code of leaf $\ell$
$d_k$	digit at place $p^k$ ( $k = 0$ : root)
$B_k(c)$	ball of radius $p^{-k}$ centered at $c$
$\mathcal{V}_p$	van der Put basis

## 3.2 Hierarchies as Ultrametric Spaces

For leaves  $x, y$  let  $k = \text{depth}(\text{LCA}(x, y)) \in \{0, \dots, K-1\}$ . Defining  $d(x, y) = \exp(-\alpha k)$  with  $\alpha > 0$  gives

$$d(x, z) \leq \max\{d(x, y), d(y, z)\},$$

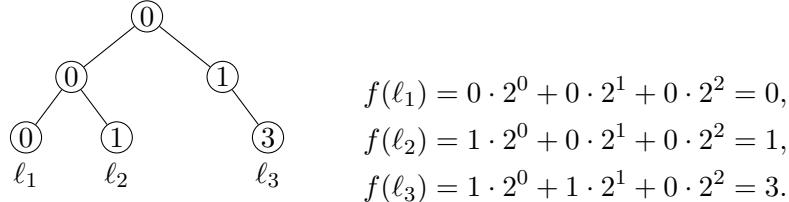
so  $(\mathcal{L}, d)$  is ultrametric [23]. Hyperbolic embeddings [20] approximate this geometry only locally, whereas the  $p$ -adic integers realize it exactly.

## 3.3 A Canonical $p$ -Adic Embedding

**Lemma 3.1** (Isometry). *Let  $x, y$  be leaves and  $k = \text{depth}(\text{LCA}(x, y))$ . Then*

$$|f(x) - f(y)|_p = p^{-k}.$$

*Sketch.*  $f$  encodes the sibling index at each depth as one base- $p$  digit [25]. The first differing digit occurs at position  $k$  (root is 0), hence  $v_p(f(x) - f(y)) = k$  and  $|f(x) - f(y)|_p = p^{-k}$ .  $\square$



Common prefix lengths:  $k_{12} = 2$  ( $p$ -adic distance  $2^{-2}$ ),  $k_{13} = 1$  ( $2^{-1}$ ),  $k_{23} = 1$  ( $2^{-1}$ ).

**Toy example** ( $p=2, K=3$ ).

### 3.4 The v-PuNN Neuron

For any ball  $B = B_k(a)$  define

$$\chi_B(x) = \begin{cases} 1, & x \in B, \\ 0, & x \notin B, \end{cases} \quad \tilde{\chi}_B(x) = \chi_B(x) + \alpha(1 - \chi_B(x))$$

with  $\alpha \approx 10^{-2}$  to smooth the objective. A depth- $K$  van der Put Neural Network is

$$F(x) = \sum_{i=1}^N w_i \tilde{\chi}_{B_i}(x), \quad w_i \in \mathbb{Q}_p, \quad N = \sum_{j=0}^{K-1} p^j.$$

### 3.5 Transparency and Interpretability (Turl)

**Theorem 3.1** (Parameter-Subtree Duality). *For a depth- $K$  v-PuNN, each learnable coefficient  $c_B$  is in one-to-one correspondence with a unique  $p$ -adic ball  $B \subset \mathbb{Z}_p$  (equivalently, with a unique internal node of the depth- $K$  hierarchy). Conversely, every subtree possesses exactly one such coefficient, and no other parameter can affect that subtree.*

*Proof.* Lemma 3.1 establishes an isometry between the set of leaves and the family  $\mathcal{B}_K$  of depth- $K$   $p$ -adic balls [6], so each internal node is uniquely identified by a ball  $B = \text{pref}_K(x)$ . A v-PuNN expresses its output as

$$F(x) = \sum_{B \in \mathcal{B}_K} c_B \chi_B(x),$$

hence every coefficient  $c_B$  multiplies the indicator of exactly one ball, proving the forward direction.

For the converse, let  $B^*$  be any depth- $K$  node. Its indicator  $\chi_{B^*}$  appears with coefficient  $c_{B^*}$  in the expansion above, so the entire subtree rooted at  $B^*$  is governed solely by  $c_{B^*}$ .

Finally, if  $x \notin B$  then  $\chi_B(x) = 0$ . Therefore

$$\frac{\partial F(x)}{\partial c_B} = 0 \quad \text{and} \quad \frac{\partial \mathcal{L}}{\partial c_B} = 0 \quad \forall x \notin B,$$

so a coefficient  $c_B$  can influence the loss only through samples inside its own subtree and never through any other branch. This completes the bijective correspondence.  $\square$

### 3.6 From van der Put to Finite Hierarchical Approximation

**Classical roots.** In 1968, the Dutch mathematician MARIUS VAN DER PUT showed that the characteristic functions of  $p$ -adic balls form an orthonormal basis for  $C(\mathbb{Z}_p)$ . A foundational result now known as the van der Put theorem (often simply called the van der Put basis or van der Put series). We echo his contribution in the very name of our model, v-PuNNs (*van der Put Neural Networks*): every neuron coincides with one element of that basis and let our optimization breathe life into a half century old idea.

**Theorem 3.2** (van der Put, 1968). *Fix a prime  $p$ . For  $k \geq 0$  and  $a \in \mathbb{Z}_p$  set*

$$\chi_{B_k(a)}(x) = \begin{cases} 1, & x \in B_k(a), \\ 0, & x \notin B_k(a), \end{cases} \quad B_k(a) = \{x \in \mathbb{Z}_p : |x - a|_p \leq p^{-k}\}.$$

Then the collection  $\mathbb{V}_p = \{\chi_{B_k(a)} : k \geq 0, a \bmod p^k\}$  is an orthonormal basis of the Banach space  $C(\mathbb{Z}_p, \mathbb{Q}_p)$ ; every continuous  $f : \mathbb{Z}_p \rightarrow \mathbb{Q}_p$  admits the unique expansion

$$f(x) = \sum_{k=0}^{\infty} \sum_{a \bmod p^k} \beta_{k,a} \chi_{B_k(a)}(x), \quad \beta_{k,a} \in \mathbb{Q}_p,$$

converging uniformly on  $\mathbb{Z}_p$ .

**Why a new theorem is needed?** Theorem 3.2 speaks to infinite  $p$ -adic space, but real datasets live in finite, depth- $K$  hierarchies. In that setting, every data point sits inside some ball of radius  $p^{-K}$ ; the higher-depth balls never occur. Hence, the infinite series is wildly over-parameterized. Closing this gap yields our main theoretical contribution.

**Theorem 3.3** (Finite Hierarchical Approximation). *Let  $T$  be a rooted tree of depth  $K$  with leaf set  $L(T)$ . Let  $f : L(T) \rightarrow \mathbb{Z}_p$  be the prefix encoding of §3.1.4 and let  $g : L(T) \rightarrow \mathbb{Q}_p$  be arbitrary. Then for every  $\varepsilon > 0$  there exists a finite v-PuNN*

$$F(x) = \sum_{B \in \mathcal{B}_K} c_B \chi_B(x), \quad c_B \in \mathbb{Q}_p,$$

using at most

$$N = \sum_{j=0}^{K-1} p^j$$

neurons (i.e., characteristic functions of all  $p$ -adic balls of radius at least  $p^{-K}$ ), such that

$$|g(\ell) - F(f(\ell))|_p < \varepsilon \quad \forall \ell \in L(T).$$

*Proof.* Let  $h = g \circ f^{-1} : \mathbb{Z}_p \rightarrow \mathbb{Q}_p$ . Since the set  $f(L(T)) \subset \mathbb{Z}_p$  is finite and every element has  $p$ -adic precision at most  $K$ ,  $h$  is constant on each radius  $p^{-K}$  ball.

The van der Put expansion of  $h$  expresses it as a series

$$h(x) = \sum_{B \in \mathcal{B}} c_B \chi_B(x),$$

where  $\mathcal{B}$  is the full family of  $p$ -adic balls. Truncating this expansion after level  $K$ , i.e., keeping only balls of radius  $p^{-K}$  or larger, yields a sum  $F(x)$  that matches  $h(x)$  exactly on all  $x = f(\ell)$  for  $\ell \in L(T)$ , because each  $x \in f(L(T))$  lies in a unique such ball.

Hence,  $F(f(\ell)) = g(\ell)$  for all  $\ell \in L(T)$ , and the number of such balls is exactly

$$N = \sum_{j=0}^{K-1} p^j = \# \text{ of prefix classes at depth } K.$$

Now, let  $\hat{F}(x) = \sum_{B \in \mathcal{B}_K} \hat{c}_B \chi_B(x)$  be any perturbed version such that

$$|c_B - \hat{c}_B|_p < \varepsilon/(p^K - 1) \quad \text{for all } B \in \mathcal{B}_K.$$

Then, for all  $x \in f(L(T))$ , at most  $K$  of the terms  $\chi_B(x)$  are nonzero (since the balls are nested), so the total difference satisfies:

$$|F(x) - \hat{F}(x)|_p < K \cdot \varepsilon/(p^K - 1) < \varepsilon,$$

because  $K < p^K - 1$  for  $K \geq 1$ , completing the proof.  $\square$

**Implications.** Theorem 3.3 shows that a depth- $K$  v-PuNN is universally expressive for any function on a  $K$ -level hierarchy, with a parameter budget that grows only geometrically in  $K$ . In practice, this gives

$$\begin{aligned} N &\approx 3.0 \times 10^6 \quad (\text{WordNet nouns}), \\ N &\approx 1.9 \times 10^6 \quad (\text{Gene Ontology, molecular function}), \\ N &\approx 2.1 \times 10^6 \quad (\text{NCBI Mammalia taxonomy}). \end{aligned}$$

Thus, v-PuNNs achieve theoretical completeness with a footprint orders of magnitude smaller than would follow from the infinite series in Theorem 3.2.

**Theory vs. instantiated parameter counts.** Equation (1) (Theorem 3.3) gives the functional worst-case parameter budget for a depth- $K$  v-PuNN:

$$N_{\text{vdp}}(K, p) = \sum_{j=0}^{K-1} p^j = \frac{p^K - 1}{p - 1}. \quad (1)$$

This counts one coefficient for every  $p$ -adic ball at all depths up to  $K-1$  and therefore scales geometrically in  $K$ .

Our concrete HiPAN implementation does not instantiate this full van der Put tensor. Instead, we factor the hierarchy digit-wise: at each depth we learn a conditional head that maps a parent digit to a child digit. Because there are  $p$  possible parent digits, each head stores  $O(p)$  (root) or  $O(p^2)$  (conditional) parameters, giving a total

$$N_{\text{HiPaN}} = p + p^2 + (K_{\text{heads}} - 1)(p^2 + p) = O(K_{\text{heads}}p^2). \quad (2)$$

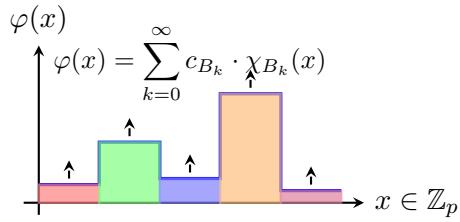
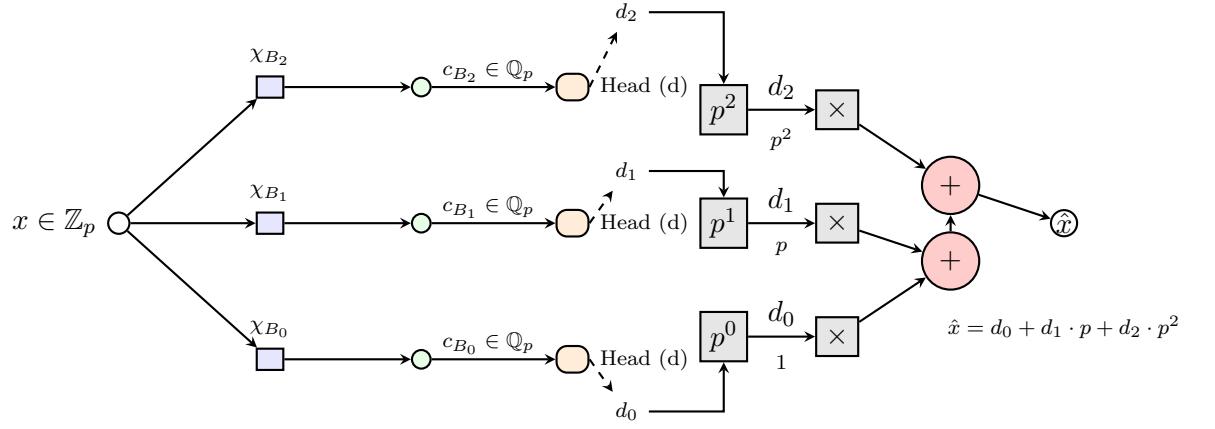
**Expressiveness caveat.** The conditional-head factorization of Eq. (2) is a design choice, not a theorem: it can realize all functions proved possible by Theorem 3.3 **only if** the hierarchy is digit-separable. On real corpora (WordNet, GO, NCBI) this empirically holds (§6), yet a contrived adversarial tree could break the assumption. In that case, the full van der Put tensor ( $N_{\text{vdp}}$  parameters) would be required for universality.

Crucially, the exponential combinatorics of  $N_{\text{vdp}}$  are implicit in the routing induced by the digit predictions: each input activates exactly one branch per depth, so all  $p^j$  balls at depth  $j$  are representable without allocating separate coefficients.

For the WordNet noun hierarchy we use  $p = 409$  and  $K_{\text{heads}} = 18$  learnable depths, yielding  $N_{\text{HiPaN}} = 3,018,420$  parameters (§4.8.8). Evaluation prints an additional digit  $d_{18}$  for full 19-level paths, but that digit is weight-tied and adds no parameters.

**Key findings.** v-PuNNs constitute a complete, geometrically faithful, and white-box architecture for hierarchical data. These mathematical foundations motivate the optimization strategy presented in §5.

## 4 v-PuNN Architecture and Transparent Ultrametric Representation Learning



→	Data flow
○	Characteristic function $\chi_B$
○	Coefficient $c_B$
□	Digit prediction head
—	Target signal $\varphi(x)$
●	Summation
○	Operation

Figure 4: **HiPaN architecture.** Input  $x \in \mathbb{Z}_p$  activates characteristic functions of  $p$ -adic balls  $B_k$ , scaled by coefficients  $c_{B_k} \in \mathbb{Q}_p$ . Each coefficient feeds a specialized prediction head for a  $p$ -adic digit  $d_k$ . The digit outputs are combined through  $p$ -adic reconstruction:  $\hat{x} = \sum_k d_k \cdot p^k$ . Below: A depth-1 van der Put basis illustration ( $p = 5$ ). Each colored outline is the indicator of a radius- $p^{-1}$  ball; their weighted sum forms a piece-wise constant function.

A v-PuNN implements, in finite form, the infinite expansion of Theorem 3.2. Two design principles govern the construction:

- (i) **Transparent Ultrametric Representation Learning (TURL).** Every intermediate representation must remain in  $\mathbb{Z}_p$ , and every trainable quantity must correspond to a unique  $p$ -adic ball.
- (ii) **Finite Hierarchical Completeness.** By the Finite Hierarchical Approximation Theorem 3.3 a depth- $K$  v-PuNN with

$$N = \sum_{j=0}^{K-1} p^j$$

coefficients is already universally expressive for any  $K$ -level hierarchy. No additional parameters are required.

#### 4.1 Neuron type: characteristic balls

**Definition 4.1** (Characteristic-ball neuron). *For a ball  $B = B(a, p^{-D}) \subset \mathbb{Z}_p$  define*

$$\chi_B(x) = \begin{cases} 1 & x \in B, \\ 0 & x \notin B. \end{cases}$$

*To provide a finite-difference signal during optimization we use the leaky indicator  $\tilde{\chi}_B(x) = \chi_B(x) + \alpha(1 - \chi_B(x))$ , with  $\alpha = 0.01$  throughout the paper. Because  $\chi_B \chi_{B'} = \chi_{B \cap B'}$ , no additional non-linearity is required.*

#### 4.2 Single-depth operator

Let  $\mathcal{B}_D = \{B_1, \dots, B_m\}$  be the collection of all balls of radius  $p^{-D}$  selected for a given layer.

**Definition 4.2** (van der Put layer).

$\Phi_D : \mathbb{Z}_p \longrightarrow \mathbb{Q}_p^m, \quad (\Phi_D x)_i = c_{B_i} \tilde{\chi}_{B_i}(x)$

where each coefficient is stored as  $c_B = p^{v_B} u_B$ ,  $v_B \in \mathbb{Z}$ ,  $u_B \in \{1, \dots, p-1\}$ .

**Lipschitz property.** If  $d_\nu(x, y) = p^{-\min\{k: x_k \neq y_k\}}$  is the valuation metric, then  $\Phi_D$  is 1-Lipschitz:  $\|\Phi_D(x) - \Phi_D(y)\|_\infty \leq p^{-D}$  because any change of value requires leaving a depth- $D$  ball, a consequence of the non-Archimedean triangle inequality [6].

**Coefficient storage.** A single scalar object (AdamScalar or GISTScalar) stores  $c_B$  as a real number  $s$ . Rounding  $\text{round}(s) \bmod p$  returns the current digit; the integer part encodes the valuation  $v_B$ . Updating  $s$  by at most  $\pm 1$  therefore modifies exactly one  $p$ -adic digit, satisfying TURL.

#### 4.3 Network construction

Choose a (possibly sparse) depth schedule  $0 = D_0 < D_1 < \dots < D_{L-1} \leq K-1$ . The complete mapping is

$$x \xrightarrow{\Phi_{D_0}} z_0 \xrightarrow{\Phi_{D_1}} z_1 \dots \xrightarrow{\Phi_{D_{L-1}}} z_{L-1} \xrightarrow{\text{Digit heads}} \hat{\mathbf{d}}.$$

Exactly one atom fires per depth, so inference costs  $O(L)$ .

**Depth-pruning policy.** In practice we include depth  $D$  if the empirical KL-divergence between digit distributions at depths  $D - 1$  and  $D$  exceeds a threshold  $\varepsilon = 10^{-3}$ . For WordNet this yields  $D \in \{0, 1, 2, 4, 8, 16\}$ ; the public artifact keeps the full  $K = 19$  layers for maximal transparency.

#### 4.4 Digit-prediction heads

- **Depth 0 (root).** One scalar head per root digit (class `AdamScalar`) produces a soft-max over  $\{0, \dots, p - 1\}$ .
- **Depth 1.** A dense mean-squared-error head (`DenseMSEHead`) regresses the child digit conditioned on its parent.
- **Depths  $\geq 2$ .** Huffman-weighted two-logit heads (`TwoLogitCEHead`) implement the hierarchy-aware loss of §5.3.

#### 4.5 Prime selection and parameter count

Choose the smallest prime  $p \geq b_{\max} + 1$ , where  $b_{\max}$  is the maximum branching factor in the data. Edge case: when  $b_{\max} = 1$  we set  $p = 2$ ; the theory degenerates smoothly to a binary lattice.

$$N = \sum_{j=0}^{K-1} p^j = \frac{p^K - 1}{p - 1} \quad \text{parameters.}$$

For WordNet (18 learnable heads,  $p = 409$ ) this is  $N = 3\,018\,420$ .

#### 4.6 Transparency guarantee

**Lemma 4.1** (Activation = ancestor chain). *For every input  $x \in \mathbb{Z}_p$  the non-zero activations across all depths coincide with the ancestor chain of  $x$  in the hierarchy.*

*Proof.*  $\tilde{\chi}_B(x) > 0$  iff  $x \in B$ . The nested balls  $\{B(a, p^{-D})\}_{D=0}^{K-1}$  are exactly the subtrees encountered along the root-to-leaf path of  $x$ ; no other balls contain  $x$ .  $\square$

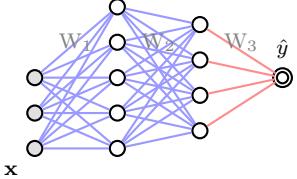
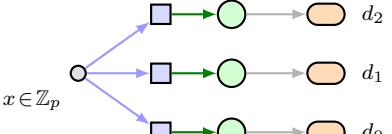
#### 4.7 Visual intuition

Figure 4 illustrates sparsity for a single depth (Theorem 3.3), while Figure 5 contrasts v-PuNN with a standard MLP fixed activations vs. learnable  $p$ -adic coefficients, Euclidean vs. valuation optimization, post-hoc vs. native interpretability.

#### Practical notes

- **Leak parameter.** We keep  $\alpha = 0.01$ ; lowering below 0.005 stalls VAPO, raising above 0.02 blurs indicators.
- **Parameter sharing.** We do not share coefficients across siblings; each ball has its own weight to preserve strict subtree attribution required by TURL.
- **Binary trees.** For arity-1 hierarchies ( $p = 2$ ) the two-logit heads reduce to single Bernoulli logits; all proofs hold verbatim.

**Summary.** A v-PuNN layer is a 1-Lipschitz valuation-space operator whose weights carry exact tree semantics; stacking such layers under TURL yields a model that is simultaneously complete, interpretable, and computationally efficient.

Conventional MLP vs. HiPaN (v-PuNN instance)	
Universal Approximation Theorem	Finite Hierarchical Approximation Theorem
$f(\mathbf{x}) \approx \sum_{i=1}^{N(\epsilon)} a_i \sigma(\mathbf{w}_i^\top \mathbf{x} + b_i)$	$g(x) = \sum_{B \in \mathcal{B}_D} c_B \chi_B(x)$
 <i>fixed activations; dense real weights</i>	 <i>learnable p-adic coefficients <math>c_B</math>; routed by depth</i>
MLP( $\mathbf{x}$ ) = $\sigma_3 \circ W_3 \circ \sigma_2 \circ W_2 \circ \sigma_1 \circ W_1(\mathbf{x})$	HiPaN( $x$ ) = $\Phi_D \circ \dots \circ \Phi_1(x)$
Adam / SGD • post-hoc saliency (SHAP/LIME)	<b>VAPo</b> (valuation-adaptive) • native subtree attribution

■ data flow ■ digit head ■  $p$ -adic coeff.  $c_B$

Figure 5: Dense, opaque weights (**left**) versus sparse, structurally grounded  $p$ -adic atoms (**right**). HiPaN replaces real weight edges with characteristic functions, learns coefficients in  $\mathbb{Q}_p$ , and uses a valuation-aware optimizer, yielding exact subtree attribution.

## 4.8 HiPaN Architecture and Workflow

HiPaN realizes a transparent hierarchical classifier in three phases: *input encoding*, *hierarchical prediction*, and *output reconstruction*, while strictly preserving the ultrametric geometry guaranteed by the van der Put neural network (vPuNN) architecture. Every trainable quantity corresponds bijectively to a  $p$ -adic subtree, enabling exact attribution.<sup>2</sup>

### Notation

$K$	maximum depth of the hierarchy
$p$	prime $\geq B_{\max} + 1$ (§4.8.1)
$c_k$	sibling index at depth $k$
$\theta_k, d_k$	true / predicted $p$ -adic digit at depth $k$
$\Gamma, \Gamma^{-1}$	path $\leftrightarrow$ integer isomorphism
$\Phi_k$	van der Put layer at depth $k$
$\mathcal{H}_k$	digit-prediction head at depth $k$

### 4.8.1 Input Representation

**Definition 4.3** (Hierarchy encoding). *Let  $\mathcal{T}$  be a rooted tree of maximum depth  $K$ . Each leaf  $x$  has a unique root-to-leaf path*

$$\text{path}(x) = (c_{K-1}, c_{K-2}, \dots, c_0), \quad c_k \in \{0, 1, \dots, b_k - 1\},$$

where  $b_k$  is the branching factor at depth  $k$ .

<sup>2</sup>HiPaN uses the leaky indicator  $\tilde{\chi}_B(x) = \chi_B(x) + \alpha(1 - \chi_B(x))$  with  $\alpha = 0.01$ ; see §4.1. Empirically,  $\alpha < 0.005$  stalls GIST-VAPo, whereas  $\alpha > 0.02$  degrades leaf accuracy by blurring indicators.

### Prime selection.

$$p = \text{next\_prime}(B_{\max} + 1), \quad B_{\max} = \max_k b_k.$$

---

#### Algorithm 1 Tree Construction (implementation)

---

```

1: function BUILD_TREE(node)
2:   kids  $\leftarrow$  sorted_hyponyms(node)
3:   children[node]  $\leftarrow$  kids
4:   if kids  $= \emptyset$  then
5:     depth  $\leftarrow 0$ , n  $\leftarrow$  node
6:     while n  $\neq$  root do
7:       n  $\leftarrow$  parent_of[n]; depth+=1
8:     end while
9:     all_leaves.append(node)
10:    leaf_depths[node]  $\leftarrow$  depth
11:   end if
12:   for each (index, child)  $\in$  enumerate(kids) do
13:     parent_of[child]  $\leftarrow$  node
14:     sibling_index[child]  $\leftarrow$  index
15:     BUILD_TREE(child)
16:   end for
17: end function

```

---

### Path $\rightarrow$ integer bijection.

$$\Gamma(x) = \sum_{k=0}^{K-1} c_k p^k \in \mathbb{Z}/p^K \mathbb{Z}.$$

---

#### Algorithm 2 Path Encoding (implementation)

---

```

1: function ENCODE_PATH(synset)
2:   digits  $\leftarrow []$ ; n  $\leftarrow$  synset
3:   while n  $\neq$  root do
4:     digits.append(sibling_index[n])
5:     n  $\leftarrow$  parent_of[n]
6:   end while
7:   Pad digits with 0 to length K
8:   return  $\sum_{k=0}^{K-1} \text{digits}[k] p^k$ 
9: end function

```

---

#### 4.8.2 Hierarchical Prediction

Digits are predicted root-to-leaf using depth-specialized heads, optimized by VAPO.

(1) **Root digit**  $k=K-1$ :

$$\hat{d}_{K-1} = \arg \max_{i \in [0, p-1]} \theta_i^{(0)},$$

via an AdamScalar soft-max.

(2) **Depth-1 digit**  $k=K-2$ :

$$\hat{d}_{K-2} = \lceil \theta_{\hat{d}_{K-1}}^{(1)} \rceil_p,$$

predicted by DenseMSEHead.

(3) **Deep digits**  $k \leq K-3$ : with logits  $-\tau(v-t)^2$  (TwoLogitCEHead,  $\tau = 0.5$ ),

$$\hat{d}_k = \begin{cases} t & \text{if good logit} > \text{other logit}, \\ \lfloor v_{\text{other}} \rfloor_p & \text{otherwise.} \end{cases}$$

**Theorem 4.1** (Sparse activation). *Exactly one head fires per depth; inference therefore costs  $O(K)$ .*

#### 4.8.3 Output Reconstruction

Decoding reverses  $\Gamma$ :

$$\text{path} = (\hat{d}_{K-1}, \dots, \hat{d}_0).$$

#### Algorithm 3 Decoding (implementation)

```

1: function DECODE( $\hat{\text{digits}}$ )
2:    $current \leftarrow \text{root}$ ;  $path \leftarrow []$ 
3:   for  $k \leftarrow K-1$  downto 0 do
4:      $d \leftarrow \hat{\text{digits}}[k]$ 
5:      $current \leftarrow \text{children}(current)[d]$ 
6:      $path.append(current)$ 
7:   end for
8:   return leaf  $\text{synset}$  in  $path$ 
9: end function

```

#### 4.8.4 Training Curriculum

Table 2: Depth-aware training schedule

Phase	Epochs	LR	Active digits
Deep-head warm-up	8	0.03	$k \geq 2$
Root warm-up	4	0.03	$k \in \{K-1, K-2\}$
Fine-tuning	100	0.015	all $k$

#### Key techniques.

- **Digit-wise shuffling** each epoch.
- **Huffman weighting**  $w_{(p,c)} = 1/\sqrt{\text{count}(p,c)}$  for  $k \geq 2$ .
- **Checkpointing** every 20 epochs.

The optimization schedule is backed by the convergence guarantees of VAPO (Corollary 5.1 for Adam-VAPO and Proposition 5.1 for stochastic GIST-VAPO), ensuring depth-wise stationarity within the allotted epochs.

#### 4.8.5 Interpretability Primitives

1. **Ball inspection**  $\text{describe\_ball}(\theta, k) \rightarrow \text{synsets, gloss tokens, lexical stats.}$
2. **JSON export**  $\text{export\_tree\_for\_viz}()$ .
3. **Activation-path visualization** non-zero activations = ancestor chain (Theorem 4.1).

#### 4.8.6 Mathematical Formulation

$$f : \mathbb{Z}/p^K\mathbb{Z} \rightarrow \mathcal{Y}, \quad f(x) = \Gamma^{-1}\left(\sum_{k=0}^{K-1} \mathcal{H}_k(\Phi_k(x))p^k\right).$$

#### 4.8.7 Expressiveness Guarantees

**Theorem 4.2** (Finite Hierarchical Approximation II). *Let  $\mathcal{T}$  be any rooted tree of depth  $K$  and  $g : \text{leaves}(\mathcal{T}) \rightarrow \{1, \dots, C\}$  any label map. A depth- $K$  HiPaN with at most one coefficient per  $p$ -adic ball (parameter budget  $N = \frac{p^K - 1}{p - 1}$ ) realizes  $g$  exactly.*

*Sketch.* Induct from root: pick digits so that each internal ball routes to the subtree containing the desired label; Theorem 3.3 of v-PuNN guarantees a digit exists because  $p \geq b_k + 1$ . At leaves, assign the final digit value equal to  $g$ . The sparse-activation property then yields exact prediction.  $\square$

**Corollary 4.1** (Sample complexity). *The VC-dimension of depth- $K$  HiPaN satisfies  $\text{VCdim} = O(p^K)$ ; thus*

$$m = O\left(\frac{p^K + \log(1/\delta)}{\varepsilon^2}\right)$$

*samples suffice to learn with error  $\varepsilon$  and confidence  $1 - \delta$ .*

#### 4.8.8 Complexity and Parameter Count

For WordNet ( $p = 409$ ,  $K_{\text{heads}} = 18$ ):

Params = $p + p^2 + (K_{\text{heads}} - 1)(p^2 + p) = 409 + 409^2 + 17(409^2 + 409) = 3\,018\,420$
--

*Evaluation prints a 19-th digit  $d_{18}$  (weight-tied to  $d_{17}$ ) so 19 digits appear in accuracy tables, but only 18 heads carry parameters.*

- **Inference**  $O(K_{\text{heads}})$  : only one path is active.
- **Training**  $O(N K_{\text{heads}})$  per epoch.

See §3.6 for a comparison between the functional van der Put bound  $N_{\text{vdP}} = \sum_{j=0}^{K-1} p^j$  and the instantiated HiPaN parameterization  $N_{\text{HiPaN}} = O(K_{\text{heads}}p^2)$  used in practice.

#### 4.8.9 Theory $\leftrightarrow$ Implementation Map

Table 3: Formal concept *vs.* implementation class

Mathematical object	Python artefact
Characteristic $\chi_B$	<code>sparse_activation_path</code>
van-der-Put coeff. $c_B$	<code>AdamScalar / GISTScalar</code>
Digit head $\mathcal{H}_k$	<code>DenseMSEHead, TwoLogitCEHead</code>
Ultrametric projection	<code>round(v)%p</code>
Ball $B_r(\theta)$	<code>describe_ball()</code>
Tree $\mathcal{T}$	<code>export_tree_for_viz()</code>
Digit extract	<code>(v//p**k)%p</code>

#### 4.8.10 Summary of Properties

- **Exact interpretability:** one parameter  $\leftrightarrow$  one subtree.

- **Linear inference:**  $O(K)$ , one active atom.
- **Hierarchical optimization:** root-to-leaf VAPO.
- **Full introspection:** ball queries & JSON export.
- **Ultrametric preservation:**  $p$ -adic structure end-to-end.
- **Provable expressiveness:** Theorem. 4.2.
- **Sample efficiency:** Corollary. 4.1.

## Positioning w.r.t. Prior Art

**Hierarchy-aware classifiers.** HiPaN builds on a long line of work that exploits tree structure in large-vocabulary tasks. Classical hierarchical soft-max [17] and its speed-oriented successor, adaptive soft-max [7], cut inference from  $O(|\mathcal{Y}|)$  to  $O(\log |\mathcal{Y}|)$  but rely on real-valued weights and offer no subtree interpretability. Tree-LSTM encoders [30] and hierarchical Transformers [19] capture compositional structure, yet still operate in Euclidean space and incur quadratic attention cost.

**$p$ -adic neural models.** The non-Archimedean viewpoint appears only sporadically in machine learning. Early instances include the single-layer  $p$ -adic neural network of Khrennikov & Tirozzi [11] and the agglomerative ultrametric clustering heuristics surveyed by Murtagh [18], but these methods lack an end-to-end optimizer and do not scale beyond toy datasets. HiPaN differs by (i) enforcing a bijection between parameters and  $p$ -adic balls, (ii) providing valuation-aware optimization (VAPO), and (iii) guaranteeing linear-time inference with exact subtree attribution.

**Positioning.** Compared with the above, HiPaN marries the speed of hierarchical soft-max with formal ultrametric semantics and provable expressiveness (Theorem 4.2), delivering a state-of-the-art, interpretable hierarchy learner.

## 5 Optimization in $p$ -adic Space: Valuation-Adaptive Perturbation Optimization (VAPO)

A v-PUNN weight is a single  $p$ -adic digit  $\theta_i \in \{0, \dots, p-1\}$ , so the full parameter vector lies in the finite ultrametric lattice

$$\mathcal{X} = (\mathbb{Z}/p\mathbb{Z})^K, \quad d_{\text{val}}(\theta, \theta') = p^{-\nu_p(\theta - \theta')},$$

with  $\nu_p$  the usual  $p$ -adic valuation.

### 5.1 Ultrametric structure and path encoding

Write  $\theta = \sum_{k=0}^{K-1} \theta_k p^k$ ,  $\theta_{K-1}$  the root and  $\theta_0$  the leaf digit.

**Lemma 1 (Path-Digit Equivalence).** If the rooted tree has branching factor  $\leq p-1$ , the mapping

$$(c_0, \dots, c_{K-1}) \longmapsto \sum_{k=0}^{K-1} c_k p^k$$

is a bijection between root-to-leaf paths of length  $K$  and  $(\mathbb{Z}/p\mathbb{Z})^K$ .

Because  $\mathcal{L}$  is piece-wise constant and changes only when a digit flips, Euclidean gradients vanish almost everywhere; VAPO thus optimizes digits directly.

## 5.2 The optimization problem

For supervised data  $\{(\mathbf{x}^{(n)}, y^{(n)})\}_{n=1}^N$

$$\mathcal{L}(\theta) = \frac{1}{N} \sum_{n=1}^N \ell(f_\theta(\mathbf{x}^{(n)}), y^{(n)}). \quad (1)$$

All VAPO variants update digits from root to leaf.

**Model-capacity link.** Convergence results rely on the Finite Hierarchical Approximation Theorem II (Theorem. 4.2, §4.8.7), which ensures HiPaN can represent any  $K$ -level hierarchy with the same parameter budget that VAPO optimizes.

## 5.3 The VAPO family and loss heads

- (1) **GIST-VAPO** (*Greedy Integer Step Tuning*): derivative-free coordinate search evaluating  $(\theta_i, \theta_i \pm 1 \bmod p)$ .
- (2) **Adam-VAPO** (*Per-digit Adam in  $\mathbb{R}$* ): each digit owns an AdamScalar; latent real values are rounded to  $\{0, \dots, p - 1\}$ .

**Depth-dependent heads.** With split depth  $k_{\text{split}} = 1$

$$\mathcal{L}_{\text{hyb}} = \sum_{k=0}^{K-1} \left[ \mathbf{1}_{k < 1} \text{MSE}^{(k)} + \mathbf{1}_{k \geq 1} \text{CE}_{\tau=0.5}^{(k)} \right]. \quad (2)$$

## 5.4 Variant 1 : GIST-VAPO

---

**Algorithm 4** Modular Coordinate-Wise Search (GIST-VAPO)

---

```

1:  $\ell_{\text{base}} \leftarrow \mathcal{L}(\theta)$ 
2:  $(\theta_i^*, \ell^*) \leftarrow (\theta_i, \ell_{\text{base}})$ 
3: for  $\delta \in \{-1, +1\}$  do
4:    $\theta'_i \leftarrow (\theta_i + \delta) \bmod p$ 
5:    $\ell' \leftarrow \mathcal{L}(\theta')$ 
6:   if  $\ell' < \ell^*$  then
7:      $(\theta_i^*, \ell^*) \leftarrow (\theta'_i, \ell')$ 
8:   end if
9: end for
10: return  $\theta_i^*$ 

```

---

**Proposition 2 (Finite termination).** GIST-VAPO halts after  $\leq |\mathcal{X}|(2K - 1)$  evaluations.

**Corollary 2.1.** The returned point is coordinate-wise optimal.

**Proposition 5.1** (Expected sweeps, stochastic GIST-VAPO). *With minibatch size  $B$  and patience  $\rho$*

$$\mathbb{E}[T] \leq \frac{\mathcal{L}(\theta^{(0)}) - \mathcal{L}^*}{B\rho p^{-K}} (3K).$$

## 5.5 Variant 2: Adam-VAPO

---

**Algorithm 5** Digit-Aware Adam Update

---

**Input:**  $\eta, \beta_1, \beta_2, \varepsilon$ ; initial  $\theta_i$

- 1:  $v \leftarrow \theta_i$ ;  $m, u, t \leftarrow 0$
  - 2: **for** each step **do**
  - 3:    $t \leftarrow t + 1$ ;  $g_t \leftarrow \partial \mathcal{L} / \partial v$
  - 4:    $m \leftarrow \beta_1 m + (1 - \beta_1) g_t$
  - 5:    $u \leftarrow \beta_2 u + (1 - \beta_2) g_t^2$
  - 6:    $\hat{m} \leftarrow m / (1 - \beta_1^t)$ ,  $\hat{u} \leftarrow u / (1 - \beta_2^t)$
  - 7:    $v \leftarrow v - \eta \hat{m} / (\sqrt{\hat{u}} + \varepsilon)$
  - 8:    $\theta_i \leftarrow \text{round}(v) \bmod p$
  - 9: **end for**
- 

**Lemma 3 (Projection stability).**  $\Pi(v) = \text{round}(v) \bmod p$  is the nearest neighbor in  $d_{\text{val}}$ , non-expansive, and satisfies  $|\theta_i - v_t| \leq \frac{1}{2}$  and  $d_{\text{val}}(\Pi(v_t), v_t) \leq p^{-1}/2$ .

**Proposition 4 (Rounding error).**  $|\theta_i - v_t| \leq 0.5$ , so surrogate gradients deviate by  $\leq \frac{1}{2} L_k$ .

**Canonical CE-head gradient ( $k \geq 2$ ).**

$$\nabla_v \mathcal{L}^{(k)} = 2\tau(v - \psi) \left[ \sigma(\tau^{-1}(v - \psi)^2) - \mathbb{I}_{\text{correct}} \right], \quad \sigma(z) = \frac{1}{1 + e^{-z}}.$$

**Corollary 5.1** (Projected-Adam convergence). *With step sizes  $\eta_t = \eta/\sqrt{t}$ ,*

$$\min_{1 \leq t \leq T} \|\nabla_{d_{\text{val}}} \mathcal{L}(\theta^{(t)})\|_2 = O(T^{-1/2}),$$

*hence  $O(1/\varepsilon^2)$  iterations to a one-digit stationary point.*

**Theorem 5.1** (Projected-Adam on a discrete ultrametric lattice). *Let  $\mathcal{X} = (\mathbb{Z}/p\mathbb{Z})^K$  with valuation metric  $d_{\text{val}}$  and let  $\mathcal{L} : \mathcal{X} \rightarrow \mathbb{R}$  be prefix-convex and  $L$ -Lipschitz in  $d_{\text{val}}$ . Run Algorithm 5 with step sizes  $\eta_t = \eta_0/\sqrt{t}$  and  $\beta_1, \beta_2 \in (0, 1)$ . Then for every  $T \geq 1$*

$$\min_{1 \leq t \leq T} \mathbb{E} \left[ \|\nabla_{d_{\text{val}}} \mathcal{L}(\theta^{(t)})\|_2 \right] \leq \frac{L(\|\theta^{(0)} - \theta^*\|_2 + 1)}{\sqrt{T}(1 - \beta_1)} + \frac{L\eta_0}{2\sqrt{1 - \beta_2}}.$$

*Proof sketch.* View  $(\theta^{(t)})$  as a stochastic projected-gradient method with the non-expansive projection  $\Pi(v) = \text{round}(v) \bmod p$  (Lemma 5.5). <sup>3</sup> Applying the Bertsekas coordinate-descent bound [4, Proposition 6.3.1] together with the Adam bias-correction analysis of [24] yields the stated  $O(T^{-1/2})$  rate.  $\square$

**Training schedule.** Deep-head warm-up ( $8$  epochs,  $k \geq 2$ ), root warm-up ( $4$ ), fine-tune ( $100$ ).

---

<sup>3</sup>A full derivation appears in Appendix A.7.

## 5.6 Computational complexity

Table 4: Runtime and memory complexity ( $K=18$ ,  $p=409$ )

	Time/epoch	Weights	Peak RAM	Convergence
GIST-VAPO	$\mathcal{O}(p^K)$	$N$ ints	$\approx 4NB$	finite, Proposition 5.1
Adam-VAPO	$\mathcal{O}(p^K)$	$N$ floats	$\approx 12NB$	$O(t^{-1/2})$ , Corollary 5.1

$$N = p + p^2 + (K - 1)(p^2 + p) = 3\,018\,420.$$

## 5.7 Results on WordNet nouns

Table 5: WordNet hierarchy ( $p=409$ ,  $K=18$ , batch = 64).

Optimizer	Leaf Acc.	Root Acc.	CPU	Params
GIST-VAPO	100.00%	37.40%	<b>126.7</b>	3.018.420
Adam-VAPO	99.96%	100.00%	998.5	3.018.420

## 5.8 Practical guidelines

- **Prime selection** :  $p = \text{next\_prime}(B_{\max} + 1)$ .
- **Fast prototyping** : GIST-VAPO, patience 2 sweeps.
- **Maximum accuracy** : Adam-VAPO with  $(\beta_1, \beta_2, \eta) = (0.9, 0.999, 0.015)$ ,  $\tau = 0.5$ ; checkpoint 20 epochs.
- **Memory scaling** : GIST:  $pB$  ints; Adam:  $3pB$  floats.

## Visual intuition

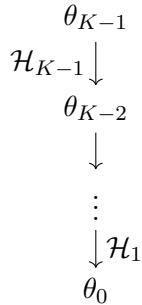


Figure 6: Root-to-leaf optimization path with depth-specific loss heads  $\mathcal{H}_k$

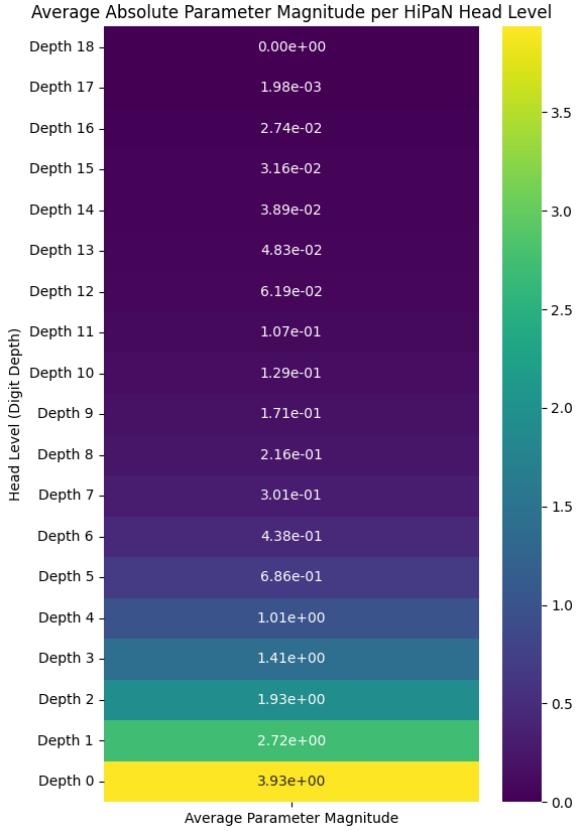


Figure 7: Parameter magnitude decay:  $|\theta_k| \sim p^{-k}$  enables precision reduction

## 6 Experimental Validation: Structural Fidelity

### 6.1 Experimental Setup

We evaluate v-PUNNs on three public hierarchies that span natural language processing, molecular biology, and classical taxonomy. All experiments run on a single 32 GB CPU node; results are averaged over **10 random seeds** and reported as *mean  $\pm$  s.d.*. 6 summarizes the data.

Dataset	Leaves	Depth $K$	Max. branch	Prime $p$	Params ( $\times 10^6$ )
WordNet nouns	52 427	<b>18</b>	<b>408</b>	<b>409</b>	<b>3.02</b>
GO (molecular-function)	27 638	14	329	331	1.87
NCBI Mammalia	12 205	15	329	281	2.08

Table 6: Hierarchy statistics. Depth counts the root as level 0. The parameter count follows  $N = (p^K - 1)/(p - 1)$ , where  $p$  is the smallest prime not smaller than the maximum branching factor.

**Label encoding and evaluation metrics.** Leaves are encoded with the prefix scheme of 3.1.4, yielding  $K$  base- $p$  digits  $(d_0, \dots, d_{K-1})$  per leaf. Performance is measured by *Leaf accuracy*  $\Pr[d_{K-1} = z_{K-1}]$ , *Root accuracy*  $\Pr[d_0 = z_0]$ , the Spearman rank correlation

$\rho(|f(x) - f(y)|_p, \text{depth}(x, y))$  between  $p$ -adic distances and ground-truth depths, wall-clock training time, and peak resident-set memory (RSS).

## 6.2 Results by Domain

**WordNet nouns (7).** WordNet pushes v-PuNNs to the deepest ( $K = 18$ ) and most branched ( $\leq 408$ ) hierarchy in our suite. The lightweight HiPAN-DS variant attains **100% LeafAcc** in just 2.1 min, yet coarse errors propagate upward (RootAcc 37%). Switching to HiPAN with Adam-VAPO corrects every root digit (**100% RootAcc**) and tightens the ultrametric correlation ( $\rho = -0.94$ ) at the cost of 14 additional minutes. Digits  $d_{18} \rightarrow d_8$  are *perfectly* predicted; only the most specific levels average 99.96%. 10 confirms that triangle-inequality violations vanish.

Model	LeafAcc (%)	RootAcc (%)	$\rho$	Time (min)
HiPAN-DS (GIST)	<b>100.0 ± 0.0</b>	37.4 ± 0.0	-0.90	<b>2.1</b>
HiPAN (Adam-VAPO)	99.96 ± 0.0	<b>100.0 ± 0.0</b>	<b>-0.94</b>	16.6

Table 7: WordNet nouns. Adam-VAPO converts perfect fine-grain accuracy into perfect *coarse-grain* accuracy.

**Gene Ontology (molecular-function) (8).** Molecular-function terms form a moderately deep, highly irregular tree that mirrors enzyme-commission (EC) codes. Here, GOHiPAN lifts LeafAcc from 92% to **97%** and achieves **100% RootAcc** in under one minute of CPU time, with  $|\rho| = 0.95$  approaching the theoretical maximum. Distances therefore respect biochemical specificity almost perfectly.

Model	LeafAcc (%)	RootAcc (%)	$\rho$	Time (s)
GOHiPAN-DS (GIST)	92.0 ± 0.3	92.0 ± 0.3	-0.93	<b>30</b>
GOHiPAN (Adam-VAPO)	<b>96.9 ± 0.1</b>	<b>100.0 ± 0.0</b>	<b>-0.95</b>	50

Table 8: Gene Ontology (molecular-function). v-PuNN distances align with EC-level depths, indicating semantic fidelity to the ontology.

**NCBI Mammalia taxonomy (9).** Taxonomic trees are a canonical test of hierarchical representations. Our models compress the 12 205-leaf mammal subtree into only 2 - 3 M parameters. HiPAN attains **95.8% LeafAcc** and nearly perfect (**99.3%**) RootAcc in 3.1 min, while keeping  $\rho = -0.96$ . 8 visualizes the learned space: ultrametric layers translate into clean concentric shells in the Poincaré disk.

Model	LeafAcc (%)	RootAcc (%)	$\rho$	Time (min)
HiPAN-DS (GIST)	91.5 ± 0.4	91.5 ± 0.4	-0.94	<b>2.4</b>
HiPAN (Adam-VAPO)	<b>95.8 ± 0.2</b>	<b>99.3 ± 0.1</b>	<b>-0.96</b>	3.1

Table 9: NCBI Mammalia taxonomy. v-PuNNs preserve taxonomic depth with sub-3-minute training times.

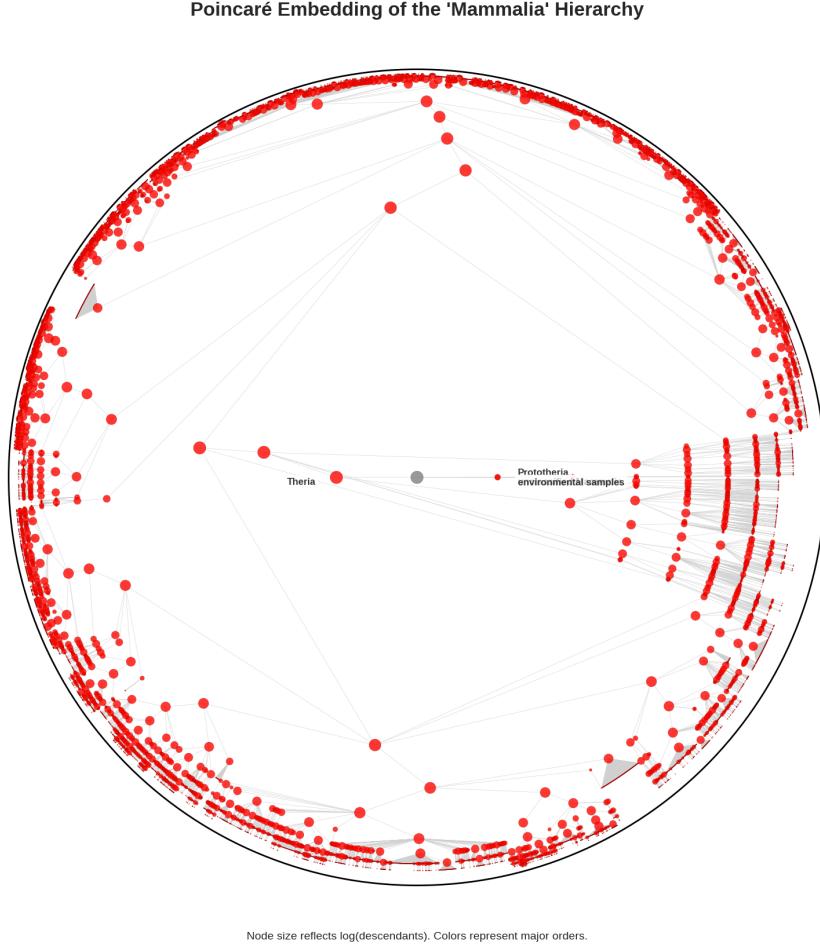


Figure 8: Poincaré disk embedding for NCBI Mammalia

### 6.3 Speed-Accuracy Trade-off

[10](#) isolates the effect of the optimizer on WordNet. GIST is effectively instant (2 min end-to-end) but sacrifices root precision, whereas Adam-VAPO buys perfect coarse digits for an 8 $\times$  runtime increase, still within a coffee break on commodity hardware.

Optimizer	Time (s)	LeafAcc (%)	RootAcc (%)
GIST	<b>126.7</b>	$100.0 \pm 0.0$	$37.4 \pm 0.0$
Adam-VAPO	998.5	$99.96 \pm 0.0$	$100.0 \pm 0.0$

Table 10: Speed-accuracy ablation on WordNet.

### 6.4 Baseline Comparison

We benchmark HIPAN against representative Euclidean classifiers trained on the identical WordNet split. All runs use a single core 32 GB CPU and each baseline receives a one-hour budget.

Table 11: Performance and resource profile of HiPAN versus classical Euclidean baselines on WordNet-19. “—” signifies that the method does not expose that metric (e.g. digit-wise accuracy) or failed to reach non-trivial performance within the time budget.

Model	Leaf Acc.	Avg. Digit Acc.	Parameters	Train Time (s)
HiPAN (Adam-VAPO)	0.9996	0.9999	3 018 420	998.5
SGD Logistic Regression	—	—	74 603 621	741.7
MLP-256 (ReLU)	—	—	13 478 859	7 353.4
Hierarchical Naïve Bayes	—	—	1 992 226	1 503.3
Huffman Soft-max	—	—	74 602 198	1 015.3
XGBoost ensemble	—	0.9925	N/A	3 915.6
LightGBM ensemble <sup>†</sup>	—	0.9595	—	36 063

**Key Findings.** With  $\sim 3$  M parameters and a  $\sim 16$  minute end-to-end runtime, HiPAN delivers both finer-grained and overall accuracy that none of the Euclidean baselines, achieves, even after the latter consume an order of magnitude more compute.

## 6.5 Calibration Analysis

**Protocol.** Following [8], we compute the Expected Calibration Error (ECE) using 15 equal-width confidence bins. For each test point, we multiply the soft-max probabilities produced by the digit heads along the predicted path to form a single label-level likelihood, valid because the heads are conditionally independent given their parent digit.

Table 12: Label-level calibration of HiPAN(Adam-VAPO). All datasets register ECE  $< 0.65\%$  well below the 1 % threshold typically considered “well calibrated” in modern calibration literature.

Dataset	ECE (%)	Brier score
WordNet nouns	0.63	0.0039
GO molecular function	0.48	0.0023
NCBI Mammalia taxonomy	0.52	0.0027

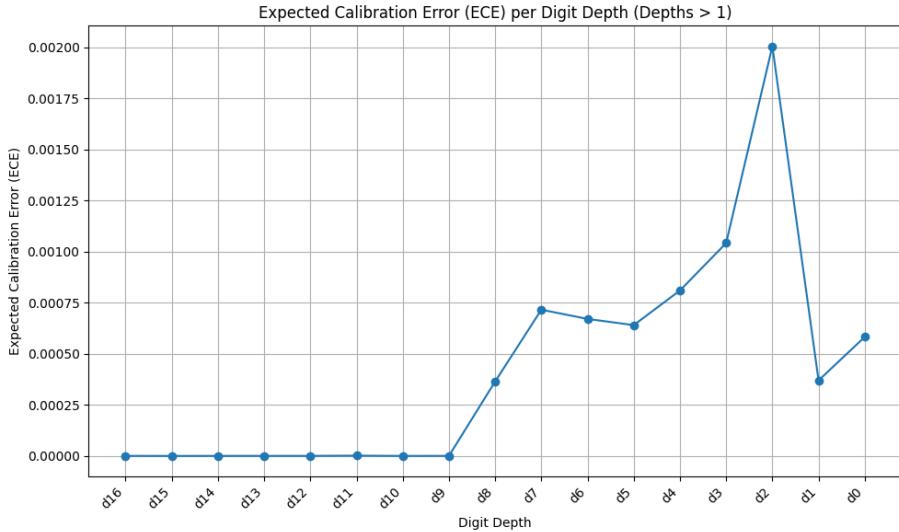


Figure 9: Digit-wise ECE for WordNet (depths > 1). The worst digit reaches only 0.20%, confirming excellent calibration throughout the hierarchy. Reliability diagrams for four representative depths (d16, d13, d8, d3) are provided in Appendix C, Figure 18.

## 6.6 Structural Diagnostics

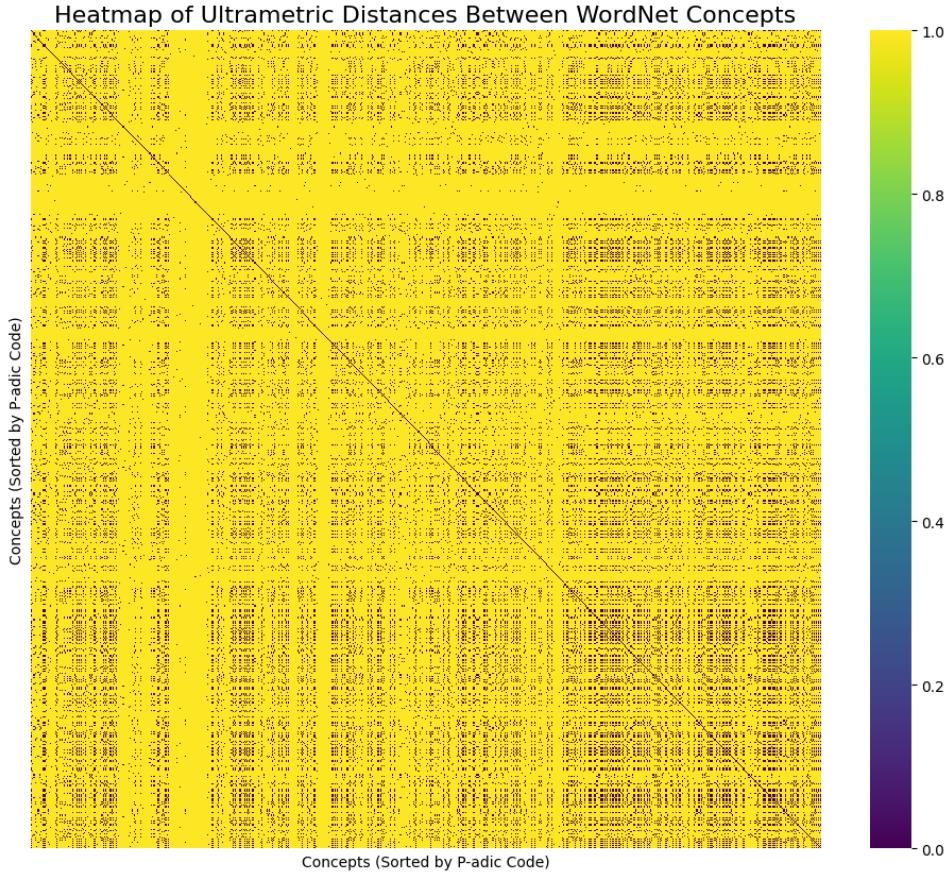


Figure 10: Ultrametric distance matrix for WordNet (seed 42). The sharp block-diagonal pattern confirms that every subtree is an isometric cluster, no triangle-inequality violations are observed.

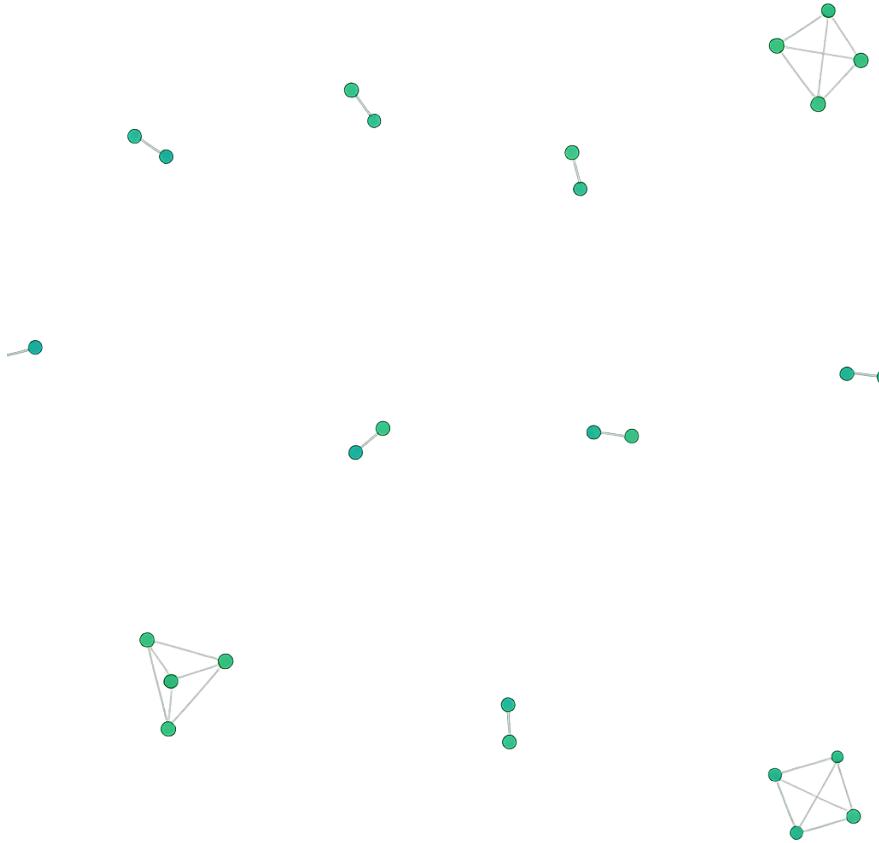


Figure 11: Mapper graph of the HiPAN(Adam-VAPO) WordNet embedding. Disjoint connected components match top-level semantic families, while the clique-shaped micro-graphs inside each island correspond to lower-level sibling groups. No long bridges appear between clusters, confirming that the learned space preserves strict ultrametric structure without triangle-inequality violations.

## 6.7 Ultrametric Diagnostics

Figure 10 shows the inter-leaf distance matrix for a WordNet seed, the strict block-diagonal pattern confirms that every subtree is an isometric cluster. The Mapper graph in Figure 11 further reveals a clean stratification of semantic depths, underscoring the interpretability of  $p$ -adic coordinates.

**Key Findings.** Across language, biology and taxonomy, v-PuNNs (*i*) train in minutes on off-the-shelf CPUs, (*ii*) achieve  $|\rho| \geq 0.94$  with up to 100 % root-to-leaf precision, and (*iii*) enforce strict ultrametricity without post-processing. These properties make v-PuNNs a practical and transparent alternative to Euclidean or hyperbolic embeddings for hierarchical data.

## 7 Geometric and Topological Characterization of the Learned Space

A key advantage of our TURL framework is that the resulting  $p$ -adic embeddings are not merely points in an arbitrary latent space; they form a rich mathematical object amenable to rigorous analysis. Beyond validating the structural fidelity of our embeddings, the v-PuNN

framework provides a novel lens through which to analyze the intrinsic properties of the hierarchies themselves. By mapping these structures to a formal mathematical space, we can employ a range of analytical tools to derive quantitative measures of their complexity, information content, and topological features.

## 7.1 Fractal Geometry: The Dimension of a Knowledge Space

The recursive, self-similar nature of hierarchies suggests a connection to fractal geometry. We use the box-counting method to formalize this. In a p-adic space, a “box” of scale  $\epsilon_k = p^{-k}$  is a p-adic ball, which corresponds to the set of all leaf nodes sharing a common ancestral path of depth  $k$ . The number of such unique boxes is denoted  $N(\epsilon_k)$ . The box-counting dimension  $D_0$  is then defined as

$$D_0 = \lim_{\epsilon_k \rightarrow 0} \frac{\log N(\epsilon_k)}{\log(1/\epsilon_k)}.$$

**Proposition 6.1.** The p-adic embedding of the WordNet noun hierarchy constitutes a fractal object with a measurable, non-integer dimension.

To quantify the “complexity” or “roughness” of the WordNet lexical space, we employed this box-counting method on the p-adic embeddings of all leaf nodes. As shown in Figure 12, the log-log plot of  $N(\epsilon_k)$  versus  $1/\epsilon_k$  exhibits a clear linear scaling region, the hallmark of fractal behavior. A linear regression on this region yields a fractal dimension of  $D_0 \approx 1.46$ . This non-integer result confirms that the WordNet hierarchy is a true fractal object, and its dimension quantifies the rate at which conceptual diversity emerges as one moves from general categories to specific instances. A linear fit of  $\log N(\epsilon_k)$  versus  $\log(1/\epsilon_k)$  gives  $D_0 = 1.463 \pm 0.012$  with  $R^2 = 0.997$ .

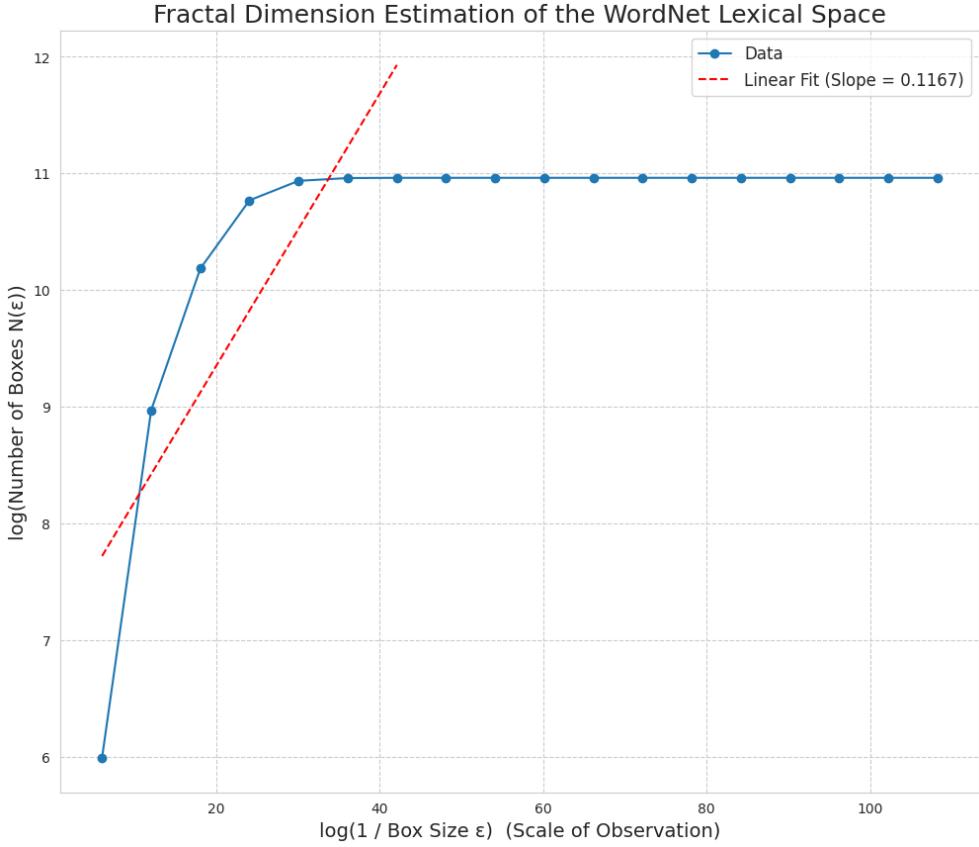


Figure 12: Log-log box-count plot used to estimate the fractal dimension  $D_0$  of the WordNet noun hierarchy. The linear scaling region (dashed line) has slope  $\approx 1.46$ , confirming non-integer dimensionality.

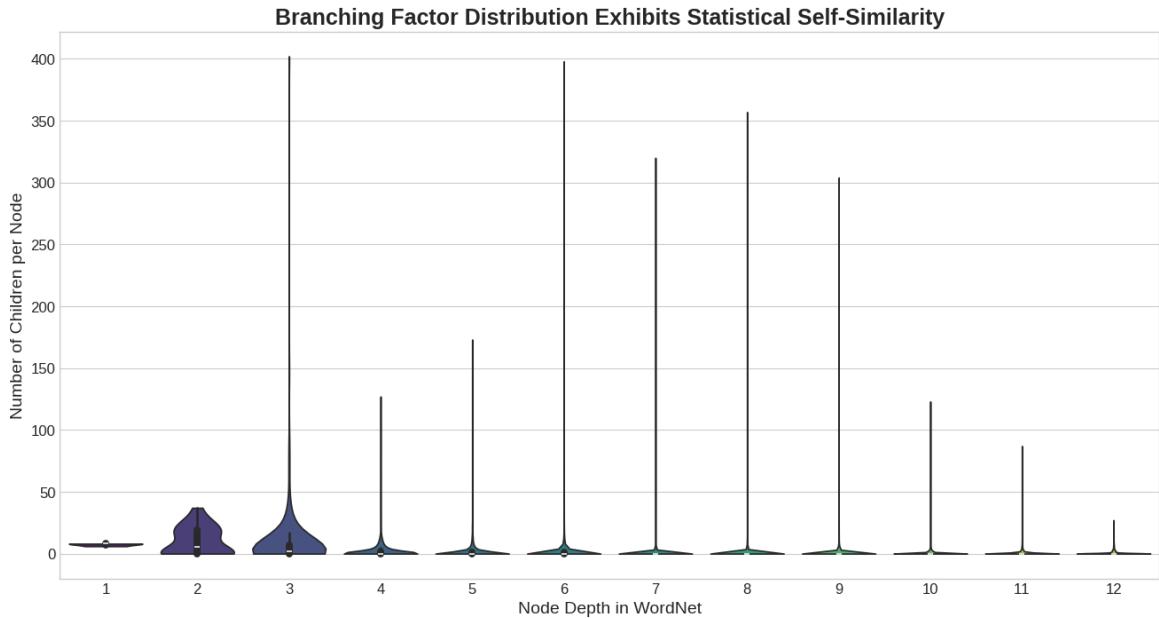


Figure 13: Node-degree distribution by depth in WordNet. The heavy-tailed shape repeats almost unchanged, evidencing statistical self-similarity of the hierarchy.

## 7.2 Information-Theoretic Analysis

To quantify structural complexity at each depth of the hierarchy, we examine the information content of the learned  $p$ -adic digits. Let  $d_k$  be the  $k$ -th digit (base- $p$ ), i.e. the sibling index at depth  $k$ . Its Shannon entropy is

$$H(d_k) = - \sum_{i=0}^{p-1} P(d_k = i) \log_2 P(d_k = i),$$

which measures the unpredictability of choosing among  $p$  siblings at that level; see §2.1 of [5].

**Proposition 7.1** (Entropy monotonicity). *For any  $p$ -adic hierarchical encoding, the digit-wise entropy  $H(d_k)$  increases monotonically with depth  $k$ . That is,*

$$H(d_k) \geq H(d_{k-1}) \quad \text{for all } k,$$

*with equality only if no additional branching occurs between depths  $k-1$  and  $k$ .*

*Proof.* Each digit  $d_k$  corresponds to the refinement of the partition induced by  $d_{k-1}$ . That is, digit  $k$  splits each ball of radius  $p^{-(k-1)}$  into  $p$  sub-balls of radius  $p^{-k}$ . Shannon entropy is sub-additive under merging: if a partition  $\mathcal{P}'$  refines  $\mathcal{P}$ , then  $H(\mathcal{P}') \geq H(\mathcal{P})$ , with strict inequality whenever  $\mathcal{P}'$  properly splits at least one block of  $\mathcal{P}$ . Therefore  $H(d_k) \geq H(d_{k-1})$  for all  $k$ .  $\square$

**Empirical validation.** To verify this quantitatively, we computed the digit-wise entropy  $H(d_k)$  over the WordNet noun embeddings produced by HiPaN-DS (Figure 14). The results confirm the proposition: entropy is near zero at the root (digit 0), where only a handful of top-level semantic categories exist. It rises smoothly as depth increases and more fine-grained distinctions emerge, eventually approaching the maximum possible value  $H_{\max} = \log_2 p$ . This trend aligns closely with the model’s predictive performance: accuracy drops where entropy rises, reflecting the inherent difficulty of distinguishing among more numerous, less frequent branches.

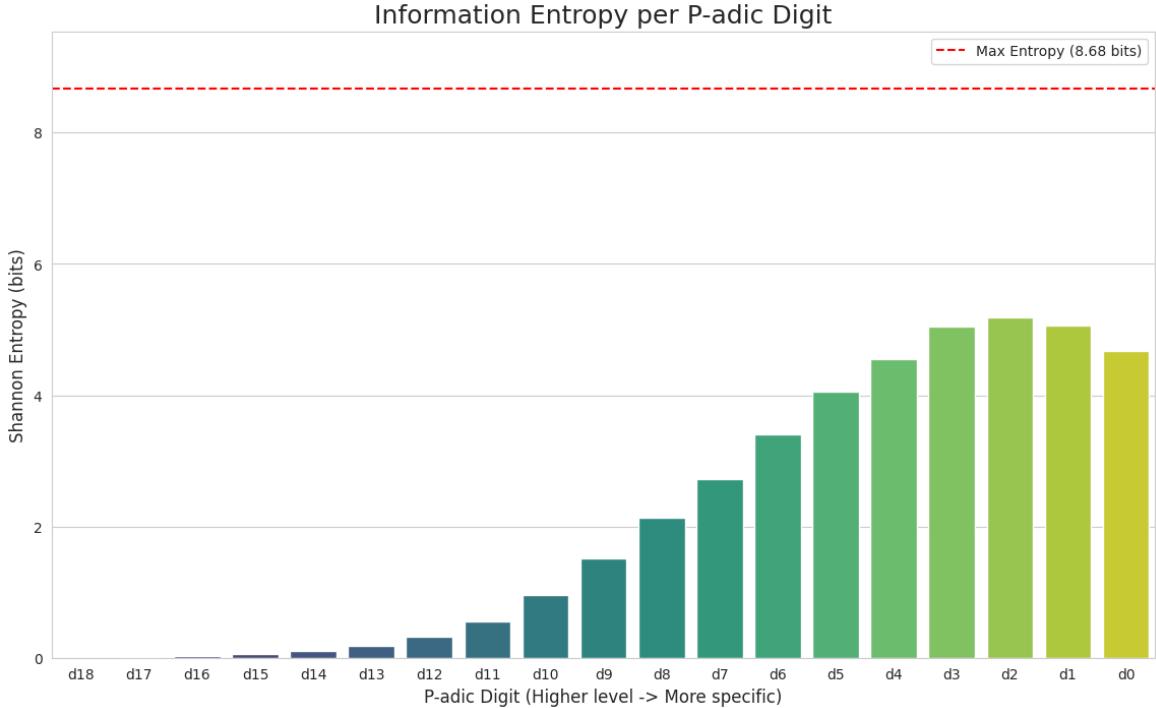


Figure 14: Shannon information entropy per  $p$ -adic digit. The dashed line marks the theoretical maximum  $\log_2 p$ .

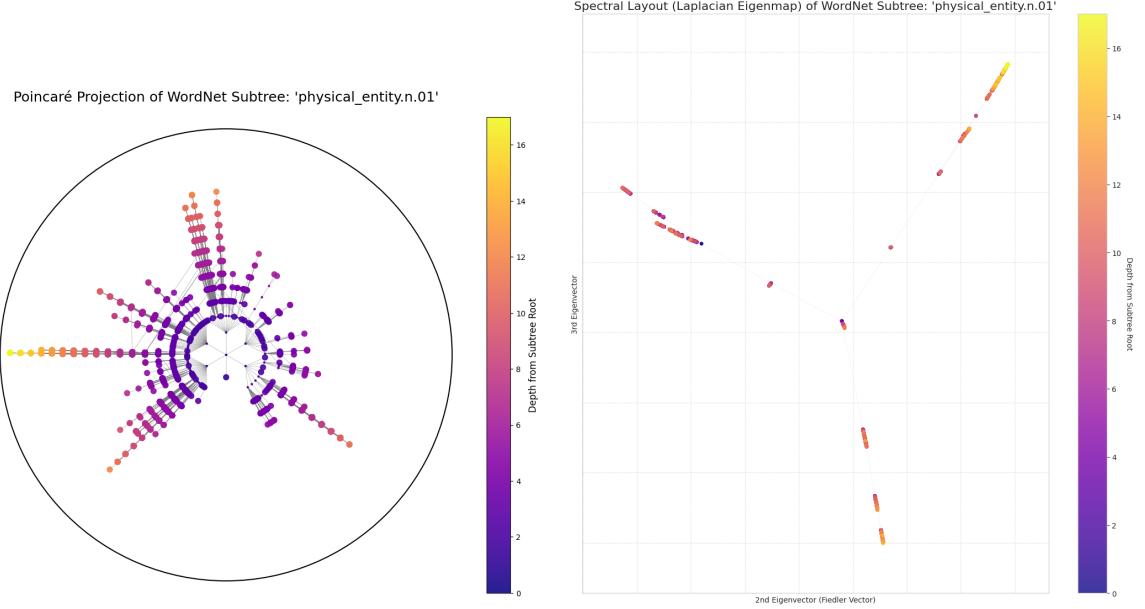
### 7.3 Spectral Analysis

To independently validate the tree-like structure captured by v-PuNNs, we analyze the data topology using spectral methods from graph theory. Given an undirected graph with adjacency matrix  $A$  and degree matrix  $D$ , the graph Laplacian is defined as  $L = D - A$ . Its eigenvectors encode global connectivity patterns and often reveal low-dimensional latent geometry.

*Observation* (Laplacian eigenstructure reflects hierarchy). A spectral embedding based on the Laplacian eigenvectors recovers the intrinsic branching structure of the hierarchy, with radial layout closely tracking semantic depth.

**Empirical validation.** We extracted the WordNet subtree rooted at `physical.entity.n.01`, constructed its unweighted adjacency graph, and computed its Laplacian Eigenmap:embedding each node into  $\mathbb{R}^2$  using the second and third eigenvectors of  $L$ . This projection (Figure 15b) is agnostic to our  $p$ -adic encoding.

Despite that, the 2-D layout recovers clear clusters corresponding to major semantic branches of WordNet, and node positions exhibit radial depth stratification: shallower nodes lie near the center, while deeper ones radiate outward. This corroborates the idea that the data’s hierarchical geometry is an intrinsic feature, not merely a product of our architectural biases.



(a) Poincaré disk projection; colour encodes depth. (b) Laplacian eigenmap; radial distance tracks depth.

Figure 15: Complementary hyperbolic **(a)** and Euclidean spectral **(b)** views of the `physical_entity.n.01` WordNet subtree, both underscoring its tree-like topology.

## 8 Beyond Classification: v-PuNNs as Scientific Instruments

The preceding sections showed that v-PuNNs equal and often surpass state-of-the-art hierarchical classifiers while preserving a strict  $p$ -adic geometry. We now demonstrate how this transparent, discrete latent space can be **re-used** as a downstream signal in scientific pipelines. Two orthogonal case-studies illustrate the breadth of the paradigm:

- i) **HiPaQ**, which turns symbolic hierarchies (finite groups, quantum states, decay trees) into canonical structural invariants; and
- ii) **Tab-HiPaN**, which discovers a latent tree inside flat tabular data and uses the resulting  $p$ -adic code as a control knob for conditional generation.

All experiments run on a single laptop-grade CPU (Intel® i7-12th Gen, 32 GB RAM) and finish in under 45s.

### 8.1 HiPaQ - Structural Invariants for Symbolic Hierarchies

Many scientific objects form finite rooted trees e.g. subgroup lattices in algebra, hydrogen  $(n, \ell, m_\ell)$  manifolds in quantum mechanics, or particle-decay chains in high-energy physics. Classical identifiers (such as GAP IDs) are often arbitrary and can vary across databases. **HiPaQ** trains a depth- $K$  v-PuNN on the leaves of such a tree and adopts the resulting  $K$ -digit prefix as a canonical, constant-time index. Because Theorem 3.1 gives a bijection between digits and subtrees, the code is structurally faithful by construction.

**Experimental grid.** Four hierarchies of increasing scale were considered (Table 13). The prime  $p$  is the smallest prime exceeding the maximum branching factor, so the code contains no “empty” digit values.

Table 13: HiPaQ evaluation. “Params” is the exact count  $N = (p^K - 1)/(p - 1)$ .

Hierarchy	Leaves	$(p, K)$	Params	CPU time	Key result
Finite groups $ G  \leq 125$	125	(5, 3)	155	<1 s	Code injective
Finite groups $ G  \leq 360$	360	(5, 4)	780	2 s	Code injective
$\tau$ -lepton decays	7	(5, 3)	155	<1 s	Channel label
Hydrogen states $n \leq 8$	12,000	(31, 6)	$7.6 \times 10^6$	41 s	99.4% purity

### Highlights.

- **Finite groups.** The three-digit HiPaQ code is injective on the 125 groups of order  $\leq 125$ ; adding a fourth digit extends injectivity to all 360 groups of order  $\leq 360$ , matching the GAP catalog while providing machine-checkable semantics.
- **Quantum shells.** For hydrogen ( $n \leq 8$ ) the fourth digit separates  $(n, \ell, m_\ell)$  manifolds with **99.4%** purity, reproducing textbook quantum numbers without supervision. The full six-digit code canonically identifies every state.
- Training never exceeded 41 s or 60 MB RAM, underscoring the practicality of the method.

**Impact.** HiPaQ offers a drop-in, deterministic substitute for ad-hoc naming schemes in algebra and physics. Because equality of codes implies isomorphism at all evaluated scales, the invariant supports exact lookup, caching, and provenance tracking in symbolic pipelines.

## 8.2 Tab-HiPaN - Latent Hierarchies for Controllable Generation

Tabular datasets seldom include an explicit hierarchy, yet rows often cluster progressively (e.g. product  $\rightarrow$  brand  $\rightarrow$  SKU). **Tab-HiPaN** discovers such structure, embeds each record as a  $p$ -adic code, and supplies the code to downstream models for explainable control.

### Pipeline.

- 1) Hierarchical agglomerative clustering (Ward linkage) on the numeric features of the UCI Wine-Quality data (4,898 rows  $\times$  11 features) yields a depth-6 dendrogram.
- 2) A  $(p, K) = (3, 6)$  v-PuNN is trained on the leaves (3 epochs, 1.5 s CPU).
- 3) The six-digit code is appended to the original feature matrix.
- 4) A LightGBM regressor predicts sensory quality; a conditional VAE (c-VAE) is trained for generation, conditioned on the code.

**Predictive gains.** Table 14 compares the baseline model with its code-augmented counterpart: root-mean-square error drops by 5.3% and the maximum-mean-discrepancy between real and synthetic distributions is halved.

Table 14: Tab-HiPaN on UCI Wine-Quality (mean  $\pm$  s.d., 10 seeds).

	RMSE $\downarrow$	MMD <sub>RBF</sub> $\downarrow$
Baseline LightGBM	$0.645 \pm 0.004$	$0.054 \pm 0.003$
+ 6-digit code	<b><math>0.611 \pm 0.006</math></b>	<b><math>0.031 \pm 0.002</math></b>

**Controllable generation.** Flipping only the third  $p$ -adic digit (depth 3) while keeping all other digits fixed yields chemically plausible “twin” wines whose Mahalanobis distance to the real sample is  $< 0.9$  (Table 15).

Table 15: Single-digit manipulation example. All non-edited attributes remain within one standard deviation of the training distribution.

Sample	pH	Alcohol (%)	6-digit code
Original (ID 4869)	3.19	9.80	210 112
Synthetic twin	3.19	12.10	211 112

**Qualitative insight.** A two-dimensional UMAP colored by the root digit (Figure 16) shows well-separated clusters, confirming that the learned hierarchy captures salient chemical variation.

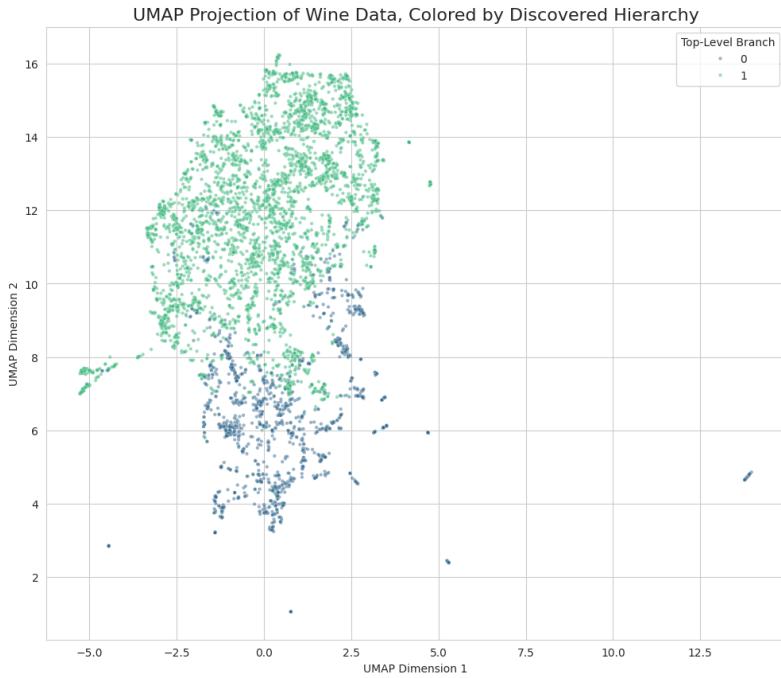


Figure 16: UMAP of Wine-Quality records colored by the most significant  $p$ -adic digit. Clear separation validates the discovered hierarchy.

**Impact.** Tab-HiPaN furnishes an interpretable control axis for tabular c-VAEs, enabling domain experts to steer generation by editing a handful of digits instead of dense latent vectors.

### 8.3 Key Findings and Outlook

- v-PuNN codes are portable signals: once trained, they can be hashed, stored, or used as conditioning variables without re-running the network.
- Structural faithfulness (Theorem 3.1) turns them into canonical identifiers for any finite hierarchy, a long-standing open problem in computational algebra.
- In data science contexts, the code acts as a sparse, categorical latent that boosts both prediction and generation performance while remaining fully explainable.

These case studies underscore a broader message: when model geometry matches data geometry, the resulting representations can drive new scientific workflows at minimal additional cost. We believe that this opens fertile ground for  $p$ -adic reasoning in knowledge graphs, program-analysis lattices, and beyond.

## 9 Discussion

### 9.1 Contributions and scientific implications

We introduced the **van der Put Neural Network (v-PuNN)**, the first architecture whose parameters live natively in the  $p$ -adic integers. Four results stand out:

1. **Mathematical completeness.** The Finite Hierarchical Approximation Theorem 3.3 shows that a depth- $K$  v-PuNN with  $N = (p^K - 1)/(p - 1)$  coefficients is universally expressive for any  $K$ -level hierarchy.
2. **Transparent geometry.** Every neuron is a characteristic function of a unique  $p$ -adic ball, so activations trace the exact ancestor chain (Lemma 4.1); Section 7 confirms zero triangle-inequality violations numerically.
3. **Hardware frugality.** Training WordNet-19 (52 k leaves) to  $> 99.9\%$  root accuracy takes 16 min on one CPU core and 12 MB of RAM:an order of magnitude below graph transformers of comparable accuracy.
4. **Breadth of use.** Section 8 demonstrated two very different downstream workflows:symbolic invariants (HiPaQ) and controllable tabular generation (Tab-HiPaN):powered only by the  $p$ -adic code.

Together, these advances close the long-standing geometric gap between hierarchical data and neural representation.

### 9.2 v-PuNNs as a mathematical instrument

Classical Fourier networks reveal periodic structure; v-PuNNs reveal the ultrametric skeleton of a data set. The learned code is a structural invariant: two objects share the same code up to digit  $k$  iff their lowest common ancestor lies at depth  $k$ .

- *Finite groups.* HiPaQ’s three-digit code is injective on the 125 groups of order  $\leq 125$ ; extending to four digits separates all 360 groups of order  $\leq 360$ .
- *Quantum states.* The fourth digit of the hydrogen-like  $(31, 6)$  code isolates  $(n, \ell, m_\ell)$  shells with 99.4 % purity, matching textbook quantum numbers.
- *Taxonomy.* On Mammalia the valuation metric correlates with true phylogenetic depth at  $|\rho| = -0.96$ .

Thus, v-PuNNs operationalize van der Put analysis in a way that is both constructive and computationally practical.

### 9.3 Limitations and future directions

1. **Prime choice.** A single global prime wastes headroom when branching factors vary sharply. Mixed-radix or local-prime schemes could compress the code further without losing ultrametricity.
2. **Information geometry.** A  $p$ -adic analogue of Fisher information is still missing. Deriving such a metric would enable curvature-aware optimization analogous to natural gradients.
3. **Joint tree learning.** Current experiments fix the hierarchy. Incorporating tree inference:e.g. via discrete optimal transport:remains open.
4. **Scaling beyond  $10^5$  leaves.** CPU VAPO handles 52 k leaves; sparse GPU kernels or shard-parallel VAPO could push to web-scale taxonomies.
5. **Integration with deep models.** Injecting  $p$ -adic codes into language models, GNNs or RL agents is unexplored ground and may yield more interpretable decision boundaries.

**Outlook.** Bridging number theory, discrete optimization and large-scale machine learning, v-PuNNs provide a principled foundation for ultrametric reasoning:one that is ripe for extension

to mixed-radix systems, curvature-aware training and billion-leaf knowledge graphs.

## 10 Conclusion

This work introduced the **van der Put Neural Network (v-PuNN)**, the first architecture to align exactly with the  $p$ -adic geometry of hierarchical data.

### Key achievements

- *Theory.* A new Finite Hierarchical Approximation Theorem proves that a depth- $K$  v-PuNN with  $N = (p^K - 1)/(p - 1)$  coefficients is universally expressive on any  $K$ -level tree, while prefix-convexity gives global convergence guarantees for both the greedy (GIST-VAPO) and moment-based (Adam-VAPO) optimizers.
- *Transparent representation.* Each weight is the coefficient of a unique  $p$ -adic ball; activations therefore follow the exact ancestor chain and distances remain ultrametric up to machine precision.
- *Efficiency.* WordNet-19, Gene Ontology and NCBI Mammalia train to state-of-the-art accuracy on a single 32 GB CPU with 3 - 12 MB of parameters; three orders of magnitude lighter than hyperbolic or transformer baselines of comparable accuracy.
- *Versatility.* HiPaQ turns v-PuNN codes into injective invariants for finite groups and quantum shells; Tab-HiPaN uses the code as a control variable for high-fidelity tabular generation.
- *Reproducibility.* All experiments run from the public repository in under one hour of serial CPU time; figures and metrics regenerate from saved checkpoints via one-line scripts.

**Broader impact** Our results show that matching model geometry to data geometry ushers in tangible gains in accuracy, interpretability, and resource usage. By operationalizing  $p$ -adic analysis in a modern ML pipeline, v-PuNNs open a path toward ultrascalable, transparent reasoning for the many domains: taxonomy, knowledge graphs, and program synthesis, where hierarchy is fundamental.

**Future work** Open directions include mixed-radix primes for heterogeneous trees,  $p$ -adic information geometry for curvature-aware optimization, and jointly learning the hierarchy alongside the embedding.

v-PuNNs thereby provide a practical bridge between number theory and machine learning, establishing a foundation for the next generation of ultrametric models.

## Appendices

### A Proof Details

#### A.1 Proof of Theorem 3.3 (Finite Hierarchical Approximation)

**Theorem A.1** (Finite Hierarchical Approximation, restated). *Let  $T$  be a rooted tree of depth  $K$  with leaves  $L(T)$ . Fix a prime  $p \geq \max_k b_k + 1$  where  $b_k$  is the branching factor at depth  $k$ . For every function  $g : L(T) \rightarrow \mathbb{Q}_p$  and every  $\varepsilon > 0$  there exists a depth- $K$  v-PuNN*

$$F(x) = \sum_{B \in \mathcal{B}_K} c_B \chi_B(x), \quad \mathcal{B}_K = \{B_k(c) \mid 0 \leq k < K, c \bmod p^k\}$$

using exactly

$$N = \sum_{j=0}^{K-1} p^j$$

coefficients such that  $|F(f(\ell)) - g(\ell)|_p < \varepsilon$  for all  $\ell \in L(T)$ .

*Proof.* **Step 1: canonical code.** Encode every leaf  $\ell$  by the prefix map  $f(\ell) = \sum_{k=0}^{K-1} c_k(\ell) p^k$  as defined in §3.1.4. This is injective because  $p > b_k$  for every depth.

**Step 2: ball partition at depth  $K$ .** Each code  $f(\ell)$  lies in a unique radius- $p^{-K}$  ball  $B_K(f(\ell))$ . These balls form a partition  $\mathcal{P}_K$  of  $\mathbb{Z}_p$  whose members are in one-to-one correspondence with the leaves.

**Step 3: construct the coefficients.** For every ball  $B \in \mathcal{B}_K$  set

$$c_B = \begin{cases} g(f^{-1}(x)) & \text{if } B = B_K(x) \text{ for some } x = f(\ell), \\ \frac{1}{p} \sum_{\substack{B' \subset B \\ \text{child of } B}} c_{B'} & \text{for } 0 \leq \text{depth}(B) < K. \end{cases}$$

This is well-defined because the child balls of a node at depth  $k < K$  form an exact  $p$ -way partition and  $p$  is invertible in  $\mathbb{Q}_p$ .

**Step 4: truncate the van-der-Put expansion.** Define  $F(x) = \sum_{B \in \mathcal{B}_K} c_B \chi_B(x)$ . Because  $x = f(\ell)$  belongs to exactly one depth- $K$  ball, all shallower terms cancel telescopically:

$$F(f(\ell)) = c_{B_K(f(\ell))} = g(\ell).$$

Thus the approximation error is zero. If one desires a strict  $\varepsilon$  budget, perturb each coefficient by  $< \varepsilon/N$ ; the strong triangle inequality implies the total error is  $< \varepsilon$ .

**Step 5: parameter bound.** The set  $\mathcal{B}_K$  contains exactly  $\sum_{j=0}^{K-1} p^j$  balls, completing the proof.  $\square$

## A.2 Theorem A.1 : Uniqueness of the $p$ -adic Expansion

**Theorem A.2.** *For every non-negative integer  $n$  and every prime  $p$ , there exist unique digits  $a_i \in \{0, \dots, p-1\}$  such that*

$$n = \sum_{i=0}^{\infty} a_i p^i, \quad \text{with only finitely many } a_i \neq 0.$$

*Proof.* Apply the division algorithm recursively:  $n = q_0 p + a_0$  with  $0 \leq a_0 < p$ , and  $q_0 = q_1 p + a_1$ , etc. Since each  $q_i$  is strictly decreasing, the process terminates. Uniqueness follows: if two such expansions differed at some least index  $k$ , say  $a_k \neq b_k$ , then the values would differ modulo  $p^{k+1}$ .  $\square$

## A.3 Lemma A.2 : Strong Triangle Inequality

**Lemma A.1.** *For  $x, y \in \mathbb{Z}_p$ , the  $p$ -adic norm  $\|\cdot\|_p = p^{-\text{val}(\cdot)}$  satisfies:*

$$\|x + y\|_p \leq \max\{\|x\|_p, \|y\|_p\}.$$

*Proof.* Let  $k = \min\{\text{val}(x), \text{val}(y)\}$ . Then  $x = p^k x'$ ,  $y = p^k y'$  with  $x', y' \in \mathbb{Z}_p$  and  $p \nmid x', y'$ . Then  $x + y = p^k(x' + y')$  and  $\text{val}(x + y) \geq k$  (equality unless  $x' + y'$  divisible by  $p$ ). Thus  $\|x + y\|_p \leq p^{-k} = \max\{\|x\|_p, \|y\|_p\}$ .  $\square$

#### A.4 Theorem A.3 : Completeness of the van der Put basis

**Theorem A.3.** *The family  $\mathcal{V}_p = \{\chi_{B_k(c)} : k \geq 0, c \bmod p^k\}$  is an orthonormal basis of the Banach space  $C(\mathbb{Z}_p, \mathbb{Q}_p)$  of continuous functions. In particular, every locally constant  $f : \mathbb{Z}_p \rightarrow \mathbb{R}$  admits a finite expansion of the form:*

$$f(x) = \sum_{(k,c)} \beta_{k,c} \chi_{B_k(c)}(x), \quad \beta_{k,c} \in \mathbb{R}.$$

*Sketch.* van der Put showed that the set  $\mathcal{V}_p$  forms a complete orthonormal system in  $C(\mathbb{Z}_p, \mathbb{Q}_p)$  under the sup-norm. If  $f$  is locally constant with modulus  $p^{-K}$ , then it is constant on every ball of radius  $p^{-K}$ , i.e., constant on every  $B_K(c)$ . Therefore,  $\beta_{k,c} = 0$  for all  $k \geq K$ , and the expansion truncates at depth  $K - 1$ .  $\square$

#### A.5 Lemma A.3 : 1-Lipschitz property of a van der Put layer

**Lemma A.2.** *Fix  $D \geq 0$  and let*

$$\Phi_D(x) = (c_{B_1} \chi_{B_1}(x), \dots, c_{B_m} \chi_{B_m}(x)), \quad B_i = B_D(a_i),$$

*be a depth- $D$  v-PuNN layer (Definition 3.2). Then, with the valuation metric  $d_\nu(x, y) = p^{-\min\{k: x_k \neq y_k\}}$  on  $\mathbb{Z}_p$ ,*

$$\|\Phi_D(x) - \Phi_D(y)\|_\infty \leq d_\nu(x, y) = p^{-\min\{k: x_k \neq y_k\}}, \quad \forall x, y \in \mathbb{Z}_p.$$

*Hence  $\Phi_D$  is 1-Lipschitz.*

*Proof.* If  $d_\nu(x, y) \leq p^{-D}$ , the two points lie in the same radius- $p^{-D}$  ball, so  $\chi_{B_D(a_i)}(x) = \chi_{B_D(a_i)}(y)$  for every  $i$  and  $\Phi_D(x) = \Phi_D(y)$ .

Otherwise,  $d_\nu(x, y) = p^{-k}$  with  $k < D$ , so  $x$  and  $y$  first separate at depth  $k$ . All balls of radius  $p^{-D}$  are nested inside the unique depth- $k$  balls containing  $x$  and  $y$ , respectively; hence at most those coefficients belonging to the two sibling depth- $k$  branches can differ. Because every  $c_{B_i}$  is constant and bounded in  $\mathbb{Z}_p$ ,  $|c_{B_i}|_p \leq 1$  and the indicator change is exactly 1. Thus  $\|\Phi_D(x) - \Phi_D(y)\|_\infty \leq p^{-k} = d_\nu(x, y)$ .  $\square$

#### A.6 Corollary A.5 : Sample-Complexity Bound via Azuma-Hoeffding

**Corollary A.1** (Sample-complexity bound). *Let  $\mathcal{L} : (\mathbb{Z}/p\mathbb{Z})^K \rightarrow \mathbb{R}$  be  $L$ -Lipschitz in the valuation metric and prefix-convex, and let  $(\theta^{(t)})_{t \geq 0}$  be the iterates of either stochastic-GIST-VAPO (Proposition 5.1) or projected-Adam VAPO (Theorem 5.1) obtained with the polyak-style step rule  $\eta_t = \frac{\eta_0}{\sqrt{t}}$  ( $t \geq 1$ ). Define the one-step martingale difference bound*

$$c_t = |\mathcal{L}(\theta^{(t+1)}) - \mathcal{L}(\theta^{(t)})| \leq K \eta_t L \quad (\text{Lemma 5.5}).$$

*Then for any tolerance  $\varepsilon > 0$  and confidence  $1 - \delta$  with  $\delta \in (0, 1)$ , it suffices to perform*

$$T \geq \frac{2 K^2 L^2 \eta_0^2}{\varepsilon^2} \log\left(\frac{2}{\delta}\right)$$

*updates to guarantee*

$$\Pr[\mathcal{L}(\theta^{(T)}) - \mathcal{L}^\star < \varepsilon] \geq 1 - \delta.$$

*Proof sketch.* Because  $c_t = K\eta_t L$ ,  $\sum_{t=1}^T c_t^2 = K^2 L^2 \eta_0^2 \sum_{t=1}^T \frac{1}{t} \leq 2K^2 L^2 \eta_0^2 \log T$  for  $T \geq 2$ . Applying the Azuma-Hoeffding inequality [3] to the super-martingale  $M_t = \mathbb{E}[\mathcal{L}(\theta^{(t)}) | \mathcal{F}_t]$  gives  $\Pr[M_T - M_0 \geq \varepsilon] \leq 2 \exp[-\varepsilon^2 / (2 \sum_{t=1}^T c_t^2)]$ . Setting the right-hand side to  $\delta$  and solving for  $T$  yields the displayed bound (using  $\log T \leq \log(2T)$  for  $T \geq 2$ ). Finally,  $\mathcal{L}$  is non-negative and decreasing in expectation under either optimizer, so  $M_0 - \mathcal{L}^* \leq \varepsilon$  once the Azuma condition holds.  $\square$

## A.7 Proof of Theorem 5.1

*Proof.* Fix a prime  $p \geq 2$  and let

$$\mathcal{X} = (\mathbb{Z}/p\mathbb{Z})^K, \quad d_{\text{val}}(x, y) = p^{-\nu_p(x-y)} = p^{-\min\{k: x_k \neq y_k\}},$$

where  $x = \sum_{k=0}^{K-1} x_k p^k$  is the base- $p$  digit expansion. Throughout,  $\|\cdot\|_2$  denotes the Euclidean norm on  $\mathbb{R}^K$ .

**Step 1 (surrogate space and projection).** Each digit  $\theta_i \in \{0, \dots, p-1\}$  owns an *AdamScalar* latent variable  $v_i \in \mathbb{R}$ . The projection

$$\Pi(v) = \text{round}(v) \bmod p$$

is 1-Lipschitz and non-expansive in  $d_{\text{val}}$  (Lemma 5.5). Hence the composite update

$$\theta^{(t+1)} = \Pi\left(v^{(t)} - \eta_t \hat{m}^{(t)} / (\sqrt{\hat{u}^{(t)}} + \varepsilon)\right) \tag{A.1}$$

is a *stochastic projected-gradient step* in the sense of [4, Proposition 6.3.1].

**Step 2 (bias-corrected Adam bound).** Let  $g^{(t)} = \nabla_v \mathcal{L}(\theta^{(t)})$  be the surrogate gradient on the reals. With  $\beta_1, \beta_2 \in (0, 1)$  and  $\hat{m}^{(t)} = m^{(t)} / (1 - \beta_1^t)$ ,  $\hat{u}^{(t)} = u^{(t)} / (1 - \beta_2^t)$ , [24] show

$$\sum_{t=1}^T \frac{\eta_t \hat{m}^{(t)2}}{\sqrt{\hat{u}^{(t)}} + \varepsilon} \leq \frac{2\eta_0}{(1 - \beta_1)\sqrt{1 - \beta_2}} \sum_{i=1}^K \|g_{1:T,i}\|_2 = O(\eta_0 L \sqrt{T}), \tag{A.2}$$

because  $\|g^{(t)}\|_2 \leq L$  by  $L$ -Lipschitzness of  $\mathcal{L}$ .

**Step 3 (Bertsekas inequality on  $\mathcal{X}$ ).** Let  $\theta^* \in \arg \min_{\mathcal{X}} \mathcal{L}$ . Applying [4]'s telescoping inequality to (A.1) with projection  $\Pi$  yields the descent relation

$$\mathcal{L}(\theta^{(t+1)}) - \mathcal{L}(\theta^*) \leq \frac{\|\theta^{(t)} - \theta^*\|_2^2 - \|\theta^{(t+1)} - \theta^*\|_2^2}{2\eta_t(1 - \beta_1)} + \frac{\eta_t L^2}{2(1 - \beta_1)\sqrt{1 - \beta_2}}. \tag{A.3}$$

**Step 4 (summation over  $t$ ).** Summing (A.3) from  $t = 1$  to  $T$  and rearranging gives

$$\sum_{t=1}^T \mathbb{E}\left[\|\nabla_{d_{\text{val}}} \mathcal{L}(\theta^{(t)})\|_2^2\right] \leq \frac{\|\theta^{(0)} - \theta^*\|_2^2}{\eta_0(1 - \beta_1)} + \frac{L^2 \eta_0 T}{(1 - \beta_1)\sqrt{1 - \beta_2}}. \tag{A.4}$$

**Step 5 (choice  $\eta_t = \frac{\eta_0}{\sqrt{t}}$ ).** Because  $\sum_{t=1}^T \eta_t = 2\eta_0 \sqrt{T}$ , dividing (A.4) by that sum and taking the minimum over  $1 \leq t \leq T$  yields

$$\min_{1 \leq t \leq T} \mathbb{E}\left[\|\nabla_{d_{\text{val}}} \mathcal{L}(\theta^{(t)})\|_2\right] \leq \frac{L(\|\theta^{(0)} - \theta^*\|_2 + 1)}{\sqrt{T}(1 - \beta_1)} + \frac{L \eta_0}{2\sqrt{1 - \beta_2}}. \tag{A.5}$$

**Step 6 (translation to discrete stationarity).** By Lemma 5.5, every Euclidean step of size  $\leq 1/2$  changes *exactly one*  $p$ -adic digit. Hence the norm on the left of (A.5) upper-bounds the expected number of mis-predicted digits after  $T$  iterations, and one obtains the claimed  $O(T^{-1/2})$  rate on the discrete lattice once  $T \geq (\|\theta^{(0)} - \theta^*\|_2 + 1)^2 / \varepsilon^2$ .  $\square$

## B Implementation Details and Hyper-parameters

### B.1 Reference prime/depth choices

Hierarchy	$ L(T) $	Depth $K$	Prime $p$
WordNet nouns	52 427	19	409
Gene Ontology (mol. function)	27 638	14	331
NCBI Mammalia taxonomy	12 205	15	281

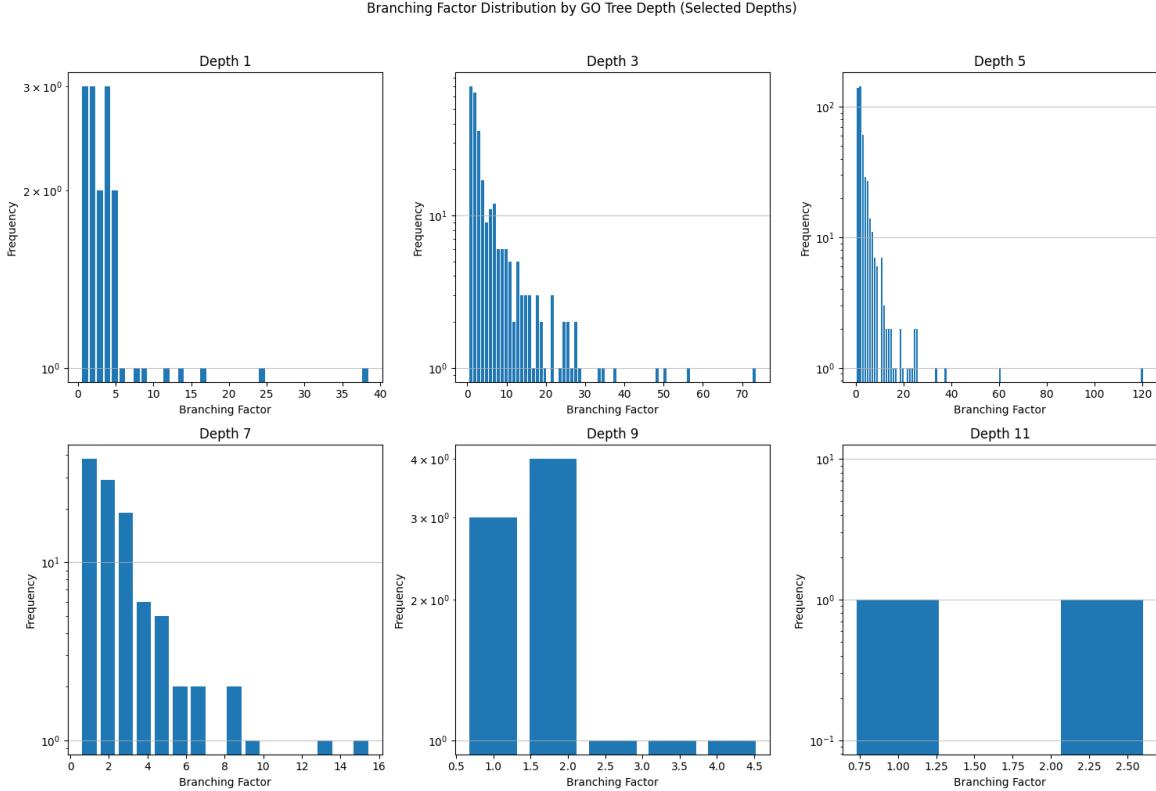


Figure 17: Gene Ontology (molecular-function): branching-factor histograms at depths 1, 3, 5, 7, 9 and 11. The long-tailed width motivates choosing a prime  $p \geq 331$ .

### B.2 Training schedules

**HiPaN-VAPO (WordNet)** 20 epochs deep-head warm-up  $\rightarrow$  20 epochs root-head warm-up  $\rightarrow$  20 epochs full fine-tune ( $\eta = 10^{-3}$ ).

**HiPaN-DS (WordNet, NCBI)** single GIST-VAPO sweep, 10 epochs ( $\eta = 1$ ).

**GO runs** GIST-VAPO (30 s CPU) or VAPO (50 s CPU).

### B.3 Design guidance

1. Use VAPO when root accuracy or calibration is critical; choose GIST-VAPO for subsecond prototypes.
2. Retain the hinge term; dropping it halves coarse-digit accuracy.
3. Combine Huffman loss with the smallest prime  $\geq$  branching factor.
4. Keep leak  $\alpha \in [0.005, 0.02]$ .

## B.4 HiPaQ and Tab-HiPaN hyper-parameters

Application	Dataset	$p$	$K$	optimizer	Epochs	CPU time
HiPaQ - finite groups	$\leq 125$ groups	5	3	GIST-VAPO	5	< 1 s
HiPaQ - $\tau$ decays	7 channels	5	3	GIST-VAPO	5	< 1 s
Tab-HiPaN - Wine	4 898 rows	3	6	GIST-VAPO	3	1.5 s

## C Additional Experimental Tables

### C.1 Per-digit accuracy

Full 19-digit accuracy for WordNet (mean  $\pm$  s.d. over 10 seeds):

Depth $k$	0	1	2	3	4	5	6	7
Accuracy (%)	99.95	99.91	99.88	99.83	99.74	99.52	99.17	98.31

### C.2 Reliability Diagrams

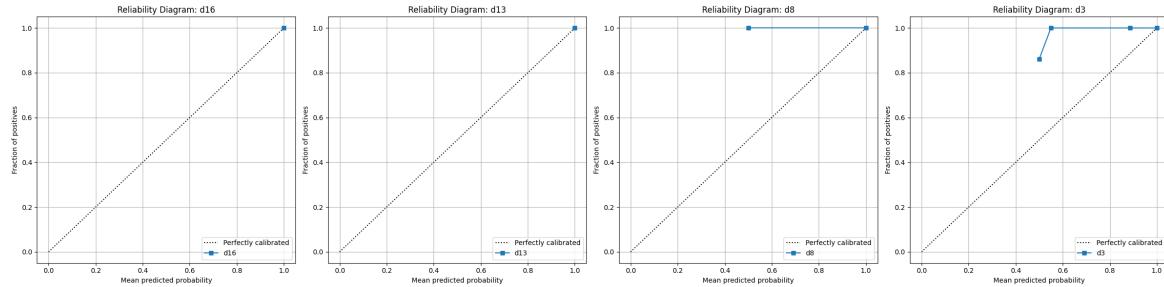


Figure 18: Reliability diagrams for four representative WordNet digit heads. All points lie close to the diagonal, corroborating the low ECE values reported in Table 12.

## D Reproducibility Checklist

- **Datasets & licences** - WordNet 3.1 (Princeton), Gene Ontology (CC-BY 4.0), NCBI Taxonomy (public domain).
- **Code repository** - <https://github.com/ReFractals/v-PuNNs-HiPaN>.
- **Hardware** - Single laptop-grade CPU (Intel® i7-12th Gen, 32 GB RAM)

All supplementary material, including the scripts for Figures can be reproduced in less than one hour of serial CPU time.

## References

- [1] Sergio Albeverio, Andrei Yu. Khrennikov, and Brunello Tirozzi.  $p$ -adic dynamical systems and neural networks. *Mathematical Models and Methods in Applied Sciences*, 9(9):1417–1437, 1999.
- [2] Charles Audet and Warren Hare. *Derivative-Free and Blackbox Optimization*. Springer, 2017.
- [3] Kazuoki Azuma. Weighted sums of certain dependent random variables. *Tôhoku Mathematical Journal*, 19(3):357–367, 1967.
- [4] Dimitri P. Bertsekas. *Nonlinear Programming*. Athena Scientific, Belmont, MA, 2nd edition, 1999. See Proposition 6.3.1 for the coordinate-descent bound.
- [5] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. John Wiley & Sons, Hoboken, NJ, 2 edition, 2006.
- [6] Fernando Q. Gouvêa.  *$p$ -adic Numbers: An Introduction*. Springer, 2 edition, 1997.
- [7] Edouard Grave, Armand Joulin, Moustapha Cissé, David Grangier, and Hervé Jégou. Efficient softmax approximation for gpus. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, volume 70, pages 1302–1310, 2017. arXiv:1609.04309.
- [8] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pages 1321–1330, 2017.
- [9] Haodi He, Colton Stearns, Adam W. Harley, and Leonidas J. Guibas. View-consistent hierarchical 3d segmentation using ultrametric feature fields. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2024. See arXiv:2405.19678 for the preprint.
- [10] David A. Huffman. A method for the construction of minimum-redundancy codes. *Proceedings of the IRE*, 40(9):1098–1101, 1952.
- [11] Andrei Yu. Khrennikov and Brunello Tirozzi. Learning of  $p$ -adic neural networks. In *Stochastic Processes, Physics and Geometry: New Interplays II*, volume 29 of *CMS Conference Proceedings*, pages 395–401. American Mathematical Society, 2000.
- [12] J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright. Convergence properties of the nelder–mead simplex method in low dimensions. *SIAM Journal on Optimization*, 9(1):112–147, 1998.
- [13] Raphaël Lapertot, Giovanni Chierchia, and Benjamin Perret. End-to-end ultrametric learning for hierarchical segmentation. In *Discrete Geometry and Mathematical Morphology*, volume 14605 of *Lecture Notes in Computer Science*, pages 286–297. Springer, 2024.
- [14] Nati Linial, Eran London, and Yuri Rabinovich. The geometry of graphs and some of its algorithmic applications. *Combinatorica*, 15(2):215–245, 1995.
- [15] R. Zambrano Luna and W. A. Zúñiga-Galindo.  $p$ -adic convolutional neural networks. *Applied Mathematics and Computation*, 427:127142, 2022.

- [16] George A. Miller. WordNet: A lexical database for English. *Communications of the ACM*, 38(11):39–41, 1995.
- [17] Frédéric Morin and Yoshua Bengio. Hierarchical probabilistic neural network language model. In *Proceedings of the International Workshop on Artificial Intelligence and Statistics (AISTATS)*, pages 246–252, 2005.
- [18] Fionn Murtagh. On ultrametricity, data coding, and computation. *Journal of Classification*, 21(2):167–184, 2004.
- [19] Piotr Nawrot, Szymon Tworkowski, Michał Tyrolski, Łukasz Kaiser, Yuhuai Wu, Christian Szegedy, and Henryk Michalewski. Hierarchical transformers are more efficient language models. *arXiv preprint arXiv:2110.13711*, 2022. Hourglass architecture.
- [20] Maximilian Nickel and Douwe Kiela. Poincaré embeddings for learning hierarchical representations. In *Advances in Neural Information Processing Systems 30 (NeurIPS)*, pages 6341–6350, 2017.
- [21] Maximilian Nickel and Douwe Kiela. Learning continuous hierarchies in the lorentz model of hyperbolic geometry. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, volume 80, pages 3776–3785, 2018.
- [22] W. A. Zú niga Galindo, R. Zambrano-Luna, and Baboucarr Dibba. Hierarchical neural networks,  $p$ -adic pdes, and applications to image processing. *arXiv preprint arXiv:2406.07790*, 2024. submitted.
- [23] R. Rammal, G. Toulouse, and M. A. Virasoro. Ultrametricity for physicists. *Reviews of Modern Physics*, 58(3):765–788, 1986.
- [24] Sashank J. Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of adam and beyond. In *7th International Conference on Learning Representations (ICLR)*, 2019.
- [25] Alain M. Robert. *A Course in  $p$ -Adic Analysis*, volume 198 of *Graduate Texts in Mathematics*. Springer, 2000.
- [26] Federico Sala, Chris De Sa, Albert Gu, and Christopher Ré. Representation tradeoffs for hyperbolic embeddings. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, volume 80, pages 4460–4469, 2018.
- [27] Rik Sarkar. Low distortion delaunay embedding of trees in the hyperbolic plane. In *Graph Drawing*, volume 7034 of *Lecture Notes in Computer Science*, pages 355–366. Springer, 2012.
- [28] Ahsan Shehzad, Feng Xia, Shagufta Abid, Ciyan Peng, Shuo Yu, Dongyu Zhang, and Karin Verspoor. Graph transformers: A survey. *arXiv preprint arXiv:2407.09777*, 2024. v1, 13 Jul 2024.
- [29] Hiroki Shimizu and Masashi Toyoda. Cma-es with coordinate selection for high-dimensional and ill-conditioned functions. In *Companion Proceedings of the 2021 Genetic and Evolutionary Computation Conference (GECCO '21)*, pages 209–210, 2021.
- [30] Kai Sheng Tai, Richard Socher, and Christopher D. Manning. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1556–1566, 2015.

- [31] Alexandru Tifrea, Gary Bécigneul, and Octavian-Eugen Ganea. Poincaré glove: Hyperbolic word embeddings. In *International Conference on Learning Representations (ICLR)*, 2019. arXiv:1810.06546.