HYPERBOLIC GENOME EMBEDDINGS

Anonymous authors
Paper under double-blind review

ABSTRACT

Current approaches to genomic sequence modeling often struggle to align the inductive biases of machine learning models with the evolutionarily-informed structure of biological systems. To this end, we formulate a novel application of hyperbolic CNNs that exploits this structure, enabling more expressive DNA sequence representations. Our strategy circumvents the need for explicit phylogenetic mapping while discerning key properties of sequences pertaining to core functional and regulatory behavior. Across 37 out of 43 genome interpretation benchmark datasets, our hyperbolic models outperform their Euclidean equivalents. Notably, our approach even surpasses state-of-the-art performance on seven GUE benchmark datasets, consistently outperforming many DNA language models while using 13–379$\times$ fewer parameters and avoiding pretraining. Our results include a novel benchmark dataset—the Transposable Elements Benchmark—which explores a significant but understudied component of the genome with deep evolutionary significance. We further motivate our work by constructing an empirical method for interpreting the hyperbolicity of dataset embeddings. Throughout these assessments, we find persistent evidence highlighting the potential of our hyperbolic framework as a robust paradigm for genome representation learning.

1 INTRODUCTION

Representation learning of genome sequence has enabled the exploration of critical unsolved problems in biology, particularly the understanding of genome function and organization (Avsec et al., 2021; Chen et al., 2022a; Dudnyk et al., 2024). Many effective approaches used for genome sequence modeling have arisen from the same machine learning methods that have powered natural language and image embeddings (Yue et al., 2023; Zhou, 2022; Consens et al., 2023). While the field has made progress by utilizing these methods, the inductive biases of these models are not usually bespoke to genomic data, limiting the expressive power of the resulting sequence representations. Given the tremendous amount of information sequestered within DNA sequences encoding cellular and molecular activity, an efficient and nuanced representation is necessary for genome interpretation and downstream analyses.

Genome organization is complex, and much of this complexity is the product of evolutionary processes. Any single genome represents the culmination of information diffusion across generations. However, this information transfer occurs through noisy channels, as background mutation rates may degrade the sequence signal (Lu et al., 2020). Accounting for phylogenetic relationships may therefore contextualize the content of the genome and ultimately benefit genome interpretation attempts. The shared influence of a common ancestor across all genomes imbues DNA sequence data with underlying hierarchical structure. These hierarchical relationships emerge through a variety of mechanisms, such as orthology and paralogy, which both codify homologous sequences but occur under different circumstances. Further compounding these interdependencies are the multiple overlapping sets of grammars for different regulatory pathways that characterize the language of the genome. Altogether, these nested levels of latent hierarchies confound genome interpretation.

In developing a modeling paradigm better suited to handling the hieararchical nature of DNA sequences, considering the geometry of the embedding spaces is essential. While most embeddings are Euclidean by default, non-Euclidean spaces may offer a compelling alternative. Specifically, hyperbolic spaces, which have the representational capacity to capture tree-structured data with high fidelity, are well-equipped to manage the hierarchical patterns ubiquitous in genomic sequences.

The negative curvature of hyperbolic spaces facilitates the continuous embedding of exponentially growing structures like phylogenetic trees with relatively low distortion.

In this work, we contend that hyperbolic spaces may be appropriate for learning meaningful representations of the genome. We leverage a fully hyperbolic framework to embed DNA sequences, implicitly handling the latent hierarchies present in the data. The main contributions of this paper are summarized as follows:

1. We adopt the machinery of fully hyperbolic convolutional neural networks (HCNNs), building two classes of HCNNs for genome sequence learning. We contrast hyperbolic and Euclidean approaches to sequence representation.
2. We introduce a novel, curated dataset–the Transposable Elements Benchmark–designed to investigate transposable elements, which remain an underexplored area of the genome with deep evolutionary roots.
3. We demonstrate the performance potential of our HCNNs across a synthetic dataset and 42 real-world datasets addressing foundational challenges in genomics.
4. We further motivate our work by formulating an empirical method for interpreting the hyperbolicity of dataset embeddings, and
5. We use this technique to interrogate properties of genome representations generated by our models, as well as from existing Euclidean models that have been widely used in the field.

2 Preliminaries

2.1 Related Work

Driven by the limitations of traditional Euclidean-based approaches in capturing relationships within complex data structures, hyperbolic deep learning methods have materialized as a promising research area. Early iterations of these methods introduced formalizations for performing the core operations of neural networks in hyperbolic space (Ganea et al., 2018; Nickel & Kiela, 2018), alongside optimization techniques generalized to Riemannian manifolds (Bécigneul & Ganea, 2019). These approaches have been further extended to a variety of frameworks, including fully hyperbolic neural networks (Chen et al., 2022b), hyperbolic graph convolutional networks (Chami et al., 2019), hyperbolic attention networks (Gulcehre et al., 2018), and hyperbolic variational auto-encoders (Mathieu et al., 2019). These models, among others, have proven effective across a variety of real-world domains, including vision (Liu et al., 2020; Hsu et al., 2021; Mathieu et al., 2019), natural language (Tifrea et al., 2019; Chen et al., 2024), and computational biology (Zhou & Sharpee, 2021; Tian et al., 2023).

In genomics, hyperbolic methods have correctly modeled established phylogenies showcasing their supremacy in representing tree-structured data (Chami et al., 2020a; Jiang et al., 2022b; Hughes et al., 2004). These methods assume that the phylogenetic tree is known *a priori*, thus the scope of the techniques are limited by the availability of evolutionary metadata. A subset of these methods produce representations of DNA sequences, but rely on an explicit mapping of phylogenetic relationships (Corso et al., 2021; Jiang et al., 2022a) in the form of pairwise edit distances or incomplete phylogenies.

2.2 Background

The n-dimensional hyperbolic space $\mathbb{H}_K^n$ is a homogeneous, simply connected Riemannian manifold, described by a constant negative curvature $K < 0$. Several equivalent formulations of hyperbolic space exist, including the Lorentz model, the Poincaré disk model, and the (Beltrami-)Klein model. Here, we use the Lorentz model, $\mathbb{L}_K^n = (\mathcal{M}^n, \mathfrak{g}_x^K)$, with manifold $\mathcal{M}^n$ and Riemannian metric $\mathfrak{g}_x^K = \text{diag}(-1, 1, \ldots, 1)$. The Lorentz model describes points by their configurations on the forward sheet of a two-sheeted hyperboloid $\mathbb{L}_K^n$ in $(n+1)$-dimensional Minkowski space. Utilizing special relativity conventions, the zeroth element in $\mathbf{x}$ is denoted as the timelike component x_t and the remaining $n-1$ elements as the spacelike components $\mathbf{x}_s$, giving $\mathbf{x} = [x_t, \mathbf{x}_s]^T$, where we can further define the timelike component $x_t = \sqrt{||\mathbf{x}_s||^2 - 1/K}$.

Exponential and logarithmic maps are used to map between the manifold $\mathcal{M}$ and tangent space $T_{\mathbf{x}}\mathcal{M}$ with $\mathbf{x} \in \mathcal{M}$. For mapping a tangent vector $\mathbf{z} \in T_{\mathbf{x}}\mathbb{L}_K^n$ onto the Lorentz manifold, we can use the

exponential map which is defined as:

$$\exp_{\mathbf{x}}^{K}(\mathbf{z}) = \cosh(\alpha)\mathbf{x} + \sinh(\alpha)\frac{\mathbf{z}}{\alpha}, \quad \text{with } \alpha = \sqrt{-K}||\mathbf{z}||_{\mathcal{L}}, \quad ||\mathbf{z}||_{\mathcal{L}} = \sqrt{\langle \mathbf{z}, \mathbf{z} \rangle_{\mathcal{L}}} \tag{1}$$

Inversely, to map a point $\mathbf{y} \in \mathbb{L}_K^n$ to the tangent space, we use the logarithmic map:

$$\log_{\mathbf{x}}^{K}(\mathbf{y}) = \frac{\cosh^{-1}(\beta)}{\sqrt{\beta^2 - 1}} \cdot (\mathbf{y} - \beta\mathbf{x}), \quad \beta = K\langle \mathbf{x}, \mathbf{y} \rangle_{\mathcal{L}} \tag{2}$$

Furthermore, in order to move points along geodesics, the parallel transport operation $\mathrm{PT}_{\mathbf{x}\to\mathbf{y}}^{K}(\mathbf{v})$ maps a point $\mathbf{v} \in T_{\mathbf{x}}\mathcal{M}$ from the tangent space of $\mathbf{x} \in \mathcal{M}$ to the tangent space of $\mathbf{y} \in \mathcal{M}$. The Lorentzian formula for parallel transport is:

$$\mathrm{PT}_{\mathbf{x}\to\mathbf{y}}^{K}(\mathbf{v}) = \mathbf{v} + \frac{\langle \mathbf{y}, \mathbf{v} \rangle_{\mathcal{L}}}{1 - K\langle \mathbf{x}, \mathbf{y} \rangle_{\mathcal{L}}}(\mathbf{x} + \mathbf{y}). \tag{3}$$

3 METHODS

3.1 FULLY HYPERBOLIC CNN

We leverage the HCNN methodology proposed by Bdeir et al. (2024) in the development of our fully hyperbolic genome sequence model. Under this framework, the elements of the traditional CNN model are reinterpreted in context of the Lorentz model of hyperbolic space. Briefly, we describe the main Lorentzian components utilized in our model.

Lorentz Convolutional Layer. In a Euclidean setting, a convolutional layer constitutes matrix mulitiplication between a linearized kernel and input feature maps. In the hyperbolic analog, each channel is defined as a separate point on the hyperboloid, with the input to each layer as an ordered set of n-dimensional hyperbolic vectors in $\mathbb{L}_K^n$. This formulation enforces the constraint that operations on points remain on the hyperboloid, as $\mathbb{L}_K^n \subset \mathbb{R}^{n+1}$. In the context of this work, each sequence is thus an ordered set of n-dimensional hyperbolic vectors, where each position describes a nucleotide in the sequence.

For a 1-dimensional hyperbolic convolutional layer with input feature map $\mathbf{x} = \{\boldsymbol{x}_l \in \mathbb{L}_K^n\}_{l=1}^{L}$, the features contained in the receptive field of kernel $\mathbf{K} \in \mathbb{R}^{m \times n \times \tilde{L}}$ are $\{\boldsymbol{x}_{l'+\epsilon\tilde{l}} \in \mathbb{L}_K^n\}_{\tilde{l}=1}^{\tilde{L}}$, in which l' marks the starting position and ϵ is the stride. Given this parameterization, we can express the convolution layer as the output of two transformations:

$$\mathbf{y}_l = \mathrm{LFC}(\mathrm{HCat}(\{\boldsymbol{x}_{l'+\epsilon\tilde{l}} \in \mathbb{L}_K^n\}_{\tilde{l}=1}^{\tilde{L}})) \tag{4}$$

Where HCat is an operation concatenating hyperbolic vectors, and LFC is a Lorentz fully-connected layer performing the affine transformation of the kernel (refer to A.1). Next, **Lorentz batch normalization** (LBN) reframes the underlying operations of batch normalization by using Fréchet mean (Lou et al., 2020) for re-centering points and Fréchet variance (Kobler et al., 2022) for re-scaling them. The algorithm is expressed as:

$$\mathrm{LBN}(\boldsymbol{x}) = \exp_{\boldsymbol{\beta}}^{K}\left(\mathrm{PT}_{\mathbf{0}\to\boldsymbol{\beta}}^{K}\left(\gamma \cdot \frac{\mathrm{PT}_{\boldsymbol{\mu}_B\to\mathbf{0}}^{K}\left(\log_{\boldsymbol{\mu}_B}^{K}(\boldsymbol{x})\right)}{\sqrt{\sigma_B^2 + \epsilon}}\right)\right). \tag{5}$$

Finally, **Lorentz multinomial logistic regression (MLR)** builds on the original formulation of a Euclidean MLR (Lebanon & Lafferty, 2004), which is defined using input $\mathbf{x} \in \mathbb{R}^n$ and C classes:

$$p(y = c|\boldsymbol{x}) \propto \exp(v_{\boldsymbol{w}_c}(\boldsymbol{x})), \quad v_{\boldsymbol{w}_c}(\boldsymbol{x}) = \mathrm{sign}(\langle \boldsymbol{w}_c, \boldsymbol{x} \rangle)\|\boldsymbol{w}_c\| d(\boldsymbol{x}, H_{\boldsymbol{w}_c}), \quad \boldsymbol{w}_c \in \mathbb{R}^n, \tag{6}$$

in which $H_{\boldsymbol{w}_c}$ is the decision hyperplane of class c. Bdeir et al. (2024) replace component operations with their Lorentzian intepretations to produce the Lorentz MLR formulation. Using parameters $a_c \in \mathbb{R}$ and $\boldsymbol{z}_c \in \mathbb{R}^n$, the Lorentz MLR's output logit for class c given input $\boldsymbol{x} \in \mathbb{L}_K^n$ is the following:

$$v_{\boldsymbol{z}_c, a_c}(\boldsymbol{x}) = \frac{1}{\sqrt{-K}}\mathrm{sign}(\alpha)\beta\left|\sinh^{-1}\left(\sqrt{-K}\frac{\alpha}{\beta}\right)\right|, \tag{7}$$

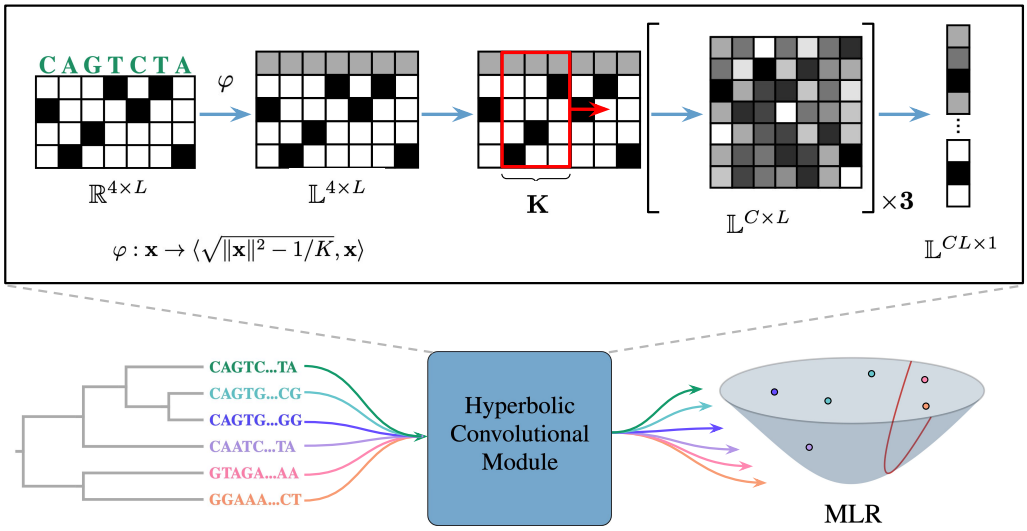

Figure 1: Overview of our HCNNs. Model inputs are sequences with latent phylogenetic structure (bottom left). As sequences pass through the hyperbolic convolutional module, they are projected onto a hyperboloid before the model convolutional and flattening steps (top insert). Using hyperbolic MLR, each sequence is classified according to the hyperplane boundaries (bottom right).

$$\alpha = \cosh(\sqrt{-K}a)\langle \boldsymbol{z}, \boldsymbol{x}_s \rangle - \sinh(\sqrt{-K}a),$$
$$\beta = \sqrt{\| \cosh(\sqrt{-K}a)\boldsymbol{z}\|^2 - (\sinh(\sqrt{-K}a)\|\boldsymbol{z}\|)^2}.$$

For further details, including Lorentz formulations of residual connections and non-linear activation, we refer the reader to Bdeir et al. (2024).

3.2 Model Overview

As our goal is to distill the difference between using Euclidean versus hyperbolic embedding spaces, we employ a relatively simple model design. The HCNN architecture consists of three major components: (1) hyperbolic convolutional blocks, (2) a flattening layer, and (3) MLR (Figure 1). Each input DNA sequence $\mathbf{x}$ is one-hot encoded at the nucleotide level, then projected channel-wise onto a hyperbolic manifold ($\varphi : \mathbb{R}^{4\times L} \rightarrow \mathbb{L}^{4\times L}$). The result of this transformation serves as the input to the hyperbolic convolutional blocks, which produce output feature maps $\mathbf{x} \in \mathbb{L}^{C\times L}$, where C is the channel dimension. After a flattening step, the model performs classification using Lorentz MLR to find the hyperbolic decision hyperplanes splitting the sequences by label.

For each hyperbolic component of our models, there exists an equivalent Euclidean component, thus we maintain architectural parity across models for a fair comparison (Appendix Figure 4). However, the layers in the HCNNs also include a learnable K parameter corresponding to the curvature of the hyperboloid on which the points reside. For our downstream experiments, we evaluate two versions of the HCNN model, HCNN-S (single K) and HCNN-M (multiple Ks). In HCNN-S, the same manifold with fixed curvature K is used across each layer of the model. In contrast, HCNN-M uses a different manifold $[K_1, ..., K_u]$ for each of u designated blocks, with intermediary steps mapping points between manifolds. By building two classes of HCNN models, we examine the trade-offs between the added representational flexibility of multiple curvatures and the potential instability introduced by incorporating multiple exponential/logarithmic mapping steps to project points onto different manifolds. Additional modeling details are in A.2.

3.3 δ-Hyperbolicity

Gromov introduces the notion of δ-hyperbolicity as a measurement of the deviation of a metric space from perfect tree-like structure (Gromov, 1987). We can define a metric space (M, d), in which the

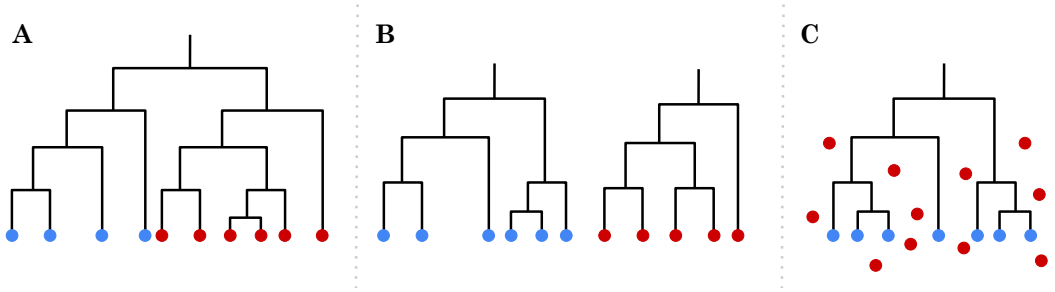

Figure 2: The various plausible evolutionary scenarios informing genomics sequence learning. Leaf coloring (blue vs. red) shows label assignment for A) intra-tree differentiation, B) inter-tree differentiation, and C) tree identification scenarios.

Gromov product of $z, y \in M$ with respect to $x \in M$ is:

$$(x, y)_z = \frac{1}{2}\left(d(x, z) + d(y, z) - d(x, y)\right) \tag{8}$$

Then, the metric space is characterized as δ-hyperbolic for some $\delta \geq 0$ if it satisfies the *four point condition* - for any four points $x, y, z, w \in M$:

$$(x, y)_w \geq \min\{(x, z)_w, (y, z)_w\} - \delta \tag{9}$$

The smallest δ for which this inequality is satisfied is the Gromov δ-hyperbolicity of (M, d).

δ-hyperbolicity has been an important tool in elucidating innate properties of metric spaces (Fournier et al., 2015; Albert et al., 2014). Recently, this measurement has been extended to explore the hyperbolic behavior of specific datasets and their respective embeddings within the domains of computer vision (Khrulkov et al., 2020), and natural language processing (Yang et al., 2024). While the original Gromov's δ (which we will denote as δ_{worst} hereinafter) is designed to represent the upper bound in terms of deviation from tree-like structure, other approaches have argued in favor of utilizing an average Gromov hyperbolicity, δ_{avg}, on the grounds that a worst case analysis of a space may not ultimately be representative of the true hyperbolic capacity of the space (Chatterjee & Sloman, 2021; Albert et al., 2014; Tifrea et al., 2019). We further develop these ideas in the context of the genomic datasets used in this paper.

As in previous approaches, we examine the behavior of δ_{worst} and δ_{avg} in high dimensional feature space. As a comparative measure, we compute a scale-invariant value of δ, which we define as $\delta_{rel} := \frac{2\delta}{D_{max}}$ (Borassi et al., 2015), where D_{max} denotes the maximal pairwise distance, or set diameter. δ_{rel} is constrained to $[0, 1]$, with a value of 0 denoting complete hyperbolicity, or perfect tree structure. Unless reported otherwise, all δs referred to in this work are the scale-invariant value.

Ultimately, both δ_{worst} and δ_{avg} are point estimates over what may be a complex landscape of δ values. To offer a more comprehensive evaluation, we examine the entire distribution of δ values across each dataset to thoroughly evaluate the hyperbolic underpinnings of DNA sequence data. By appraising the full landscape of δ-hyperbolicity in our embedding space, we gain a richer understanding of the intrinsic tree structure across each dataset. We provide further details on δ computation and other experimental configurations in A.9.1.

4 Data

4.1 Benchmarks

Synthetic Datasets. In order to rigorously interrogate the applicability of using a hyperbolic architecture in a genomics application, we create several synthetic datasets to illuminate the underlying biological processes being captured by our models. We consider the various plausible data-generating processes for biological sequences, and define three potential cases of biological signal transmission being learned by the models. Additionally, given prior evidence in Corso et al. (2021) that purely artificial sequences may not always be indicative of performance on real-world datasets, we explore

this phenomenon as well, by creating two sets of data for each case: one in which the sequences used are completely randomly generated, and one in which the sequences are randomly sampled from existing genomes.

Our synthetic datasets mimic evolutionary dynamics by perturbing input sequences based on phylogenetic tree structure. We simulate sequence evolution along tree branches with the generalized time-reversible (GTR) nucleotide model (Tavaré, 1984). We define each scenario, visualized in Figure 2, as follows:

(A) **Intra-tree differentiation**: sequences are generated from a single phylogenetic tree, with labels derived from clade membership.
(B) **Inter-tree differentiation**: sequences are generated from different phylogenetic trees, with labels derived from phylogeny membership.
(C) **Tree identification**: sequences are labeled based on the generating process: phylogenetic tree generation or non-phylogenetic (random) generation.

We leverage these scenarios to better understand the specific advantages of hyperbolic models and identify the conditions under which they demonstrate the greatest effectiveness. For full details regarding the generation of each dataset, see A.6.

Transposable Elements Benchmark. We introduce a multi-species benchmark for exploring how transposable elements are codified in sequence. Transposable elements (TEs) are highly abundant, mobile elements of genomic sequence that represent specific evolutionary trajectories within organisms (Hayward & Gilbert, 2022; Wells & Feschotte, 2020). Given their ability to move within genomes, TEs drive genomic plasticity and have been identified as key players in the evolution of genomic complexity (Schrader & Schmitz, 2019; Bowen & Jordan, 2002). TEs can influence gene expression and regulation by acting as alternative promoters (Faulkner et al., 2009), providing transcription factor binding sites (Sundaram et al., 2014), introducing alternative splicing (Shen et al., 2011), and mediating epigenetic modifications (Drongitis et al., 2019). As such, TEs have been also implicated in disease pathogenesis (J"onsson et al., 2020; Hancks & Kazazian, 2016). Overall, TEs represent a powerful force in evolutionary biology, continually shaping the genetic landscape.

A variety of TEs exist across genomes and can be arranged into several sub-classes. The genetic structure of TE types follow regular patterns of structural features and motifs, and thus represent an interesting learning opportunity for sequence models. The Transposable Elements Benchmark (TEB) presents a novel resource for investigating TEs, which represent an area of genome organization that is under-explored in the genomics deep learning literature. TEB surveys several different TE classes across plant and human genomes. Specifically, TEB offers binary classification datasets for identifying seven specific elements across three different TE classes: retrotransposons, DNA transposons, and pseudogenes. Detailed data preprocessing and statistics of each dataset in TEB are further presented in A.3.

Genome Understanding Evaluation. The Genome Understanding Evaluation (GUE) benchmark is a recently published tool that contains seven biologically significant genome analysis tasks that span 28 datasets. Designed to scrutinize the capabilities of genome foundation models, GUE prioritizes genomic datasets that are challenging enough to discern differences between models. The datasets are comprised of sequences ranging from 70–1000 base pairs in length and originating from yeast, mouse, human, and virus genomes. Further details can be found in Zhou et al. (2024).

Genomic Benchmarks. We utilize the Genomic Benchmarks (GB) resource, which consists of 8 separate classification datasets that spotlight regulatory elements across three different model organisms: human, mouse, and roundworm. Datasets were carefully constructed from published data repositories and consist of input sequences of length 200–500, with the exception of the drosophila enhancers stark dataset, in which sequences have a median length of 2,142. Full details on data preprocessing and dataset summary statistics can be found in Grešová et al. (2023). As the human non-tata promoters dataset in GB was created using data that was also used in the creation of the promoter detection datasets in GUE (Dreos et al., 2013), we note this when discussing model performance.

5 EXPERIMENTS

5.1 GENOMIC CLASSIFICATION

Classification Tasks. The results from the the three classification benchmarks and synthetic dataset are summarized in Table 1. Across the 43 distinct datasets, the hyperbolic models outperform the equivalent Euclidean model on 37 tasks, as measured by Matthew's correlation coefficient (MCC). In 29 of these datasets, this improvement in score by a hyperbolic model is statistically significant when accounting for variance across different model initializations, whereas the Euclidean CNN statistically outperforms HCNN in only two datasets.

Further examination of the results suggests that HCNNs confer a particularly strong advantage in distinguishing transcription factors binding sites (across species) and epigenetic marks, as well as in distinguishing TEs in sequence. Across promoter detection tasks, there appears to be no added benefit of a hyperbolic embedding. Since promoters likely function through more complex, combinatorial interactions, these latent hierarchies may be more challenging for HCNNs to effectively capture. HCNNs also seem to be hugely disadvantaged in the Covid variant prediction task, in distinguishing nine different variants of Covid from sequence.

Notably, when comparing the best scoring model across runs, HCNNs outperform DNA language models (LMs) in seven of the 28 GUE datasets (A.4 and Appendix Table 5). Across the majority of tasks, HCNNs outpace DNABERT (5-mer), DNABERT (6-mer), NT-500M human, NT-500M-1000g, and NT-25000M-1000g (Lopez et al., 2023). Considering the immense scale of these LMs, with $13\times$ to $379\times$ more trainable parameters than HCNNs, along with pretraining on the entire human genome and 1000 Genomes Project sequences, the performance gap is particularly striking. HCNNs appear to have a consistent advantage over Euclidean models across many of the core deep learning genomics tasks.

Expressive Power. In directly comparing the embeddings and decision boundaries learned by each class of model, we can begin to infer their differences in expressiveness. Figure 3 visualizes the distinctive class boundaries and sequence relationships learned by HCNNs and CNNs. We observe far better separation of classes in the hyperbolic embeddings than in the Euclidean case, lending further credence to the appropriateness of hyperbolic embeddings in a genomic setting.

Embedding Dimensionality. Prior work on HNNs has demonstrated that the effectiveness of hyperbolic embeddings is especially pronounced at lower dimensions (Chami et al., 2020b; Chamberlain et al., 2017). We attempted to replicate these findings under our study conditions by varying the number of channels in the convolutional blocks in both the CNNs and HCNNs. We then train and evaluate each of these distinct models on TEB.

The results in Appendix Figure 5 show that HCNN-S appears to steadily improve its advantage over the CNN at lower channel dimensions, consistent with the pattern shown in literature. At very low dimensions, the average improvement in performance is greater than at higher dimensions. However, HCNN-M does not show increased performance at lower dimensions. As HCNN-M is a more complex model compared to HCNN-S, it may be possible than a minimum model capacity is necessary before the benefits of multi-curvature representations become useful.

Learned Curvature. The curvature of the hyperbolic manifold is a learnable parameter. Exploration of this parameter in TEB (detailed in A.5 and Figure 6) illustrates that the value of K does not vary far from its default initialization value of -1. However, the HCNN-S models and HCNN-M models gravitate towards different curvature values ($K > -1$ and $K < -1$, respectively), and there are small adjustments in the curvature of the embedding spaces for each block of the model.

Hybrid Models. We construct hybrid models with mixed Lorentzian and Euclidean components (see A.8 for details). Our results indicate that Euclidean embeddings may still benefit from hyperbolic decision boundaries.

5.2 δ-HYPERBOLICITY ESTIMATION

As presented in Figure 11, our investigation reveals several notable characteristics of δ-hyperbolicity values in finite datasets. The δ (Figure 11) and δ_{worst} (Appendix Table 8) values computed from the final embedding layer are ostensibly hyperbolic; all values are closer to 0 than 1, indicating tree-like

Table 1: Model performance (MCC) on all real-world genomics datasets averaged over 5 random seeds (mean $\pm$ standard deviation). The highest scoring model is in bold, while † denotes that the hyperbolic model outperformed the Euclidean model, or that the Euclidean model outperformed the higher-scoring hyperbolic model with $p < 0.05$, Wilcoxon rank-sum test. *We note that the human non-tata promoters dataset in GB overlaps with the GUE Promoter Detection datasets.

Benchmark	Task	Dataset	Model: Euclidean CNN	Model: Hyperbolic HCNN-S	Model: Hyperbolic HCNN-M
TEB	Retrotransposons	LTR Copia	54.73±1.45	64.58±3.07 †	**68.05**±2.80 †
		LINEs	70.63±1.24	76.12±2.16 †	**77.10**±2.92 †
		SINEs	85.15±1.64	**85.45**±1.16	81.85±2.95
	DNA transposons	CMC-EnSpm	72.18±0.32	**80.98**±1.48 †	80.65±1.30 †
		hAT-Ac	87.45±0.90	89.61±1.34	**91.04**±1.58 †
	Pseudogenes	processed	60.66±0.82	**68.30±0.93** †	65.41±5.54
		unprocessed	51.94±2.69	56.13±0.56 †	**58.36**±1.80 †
GUE	Epigenetic Marks Prediction	H3	64.83±2.17	68.14±1.44	**68.32**±2.12 †
		H3K14ac	34.27±6.14	**50.37**±8.14 †	45.69±1.95 †
		H3K36me3	43.74±2.32	**53.28**±1.94 †	43.41±2.00
		H3K4me1	28.76±3.00	**40.84**±1.18 †	34.71±3.70
		H3K4me2	25.38±5.40	**39.74**±4.61 †	29.53±1.97
		H3K4me3	21.77±5.58	**49.51**±0.96 †	30.39±3.32 †
		H3K79me3	54.88±2.09	**62.39**±2.14 †	58.48±1.88
		H3K9ac	40.37±3.89	**52.90**±1.12 †	50.21±1.52 †
		H4ac	31.59±8.45	**52.29**±0.93 †	44.88±4.70
		H4	74.81±0.92	75.43±1.49	**76.20**±0.61
	Human Transcription Factor Prediction	0	58.65±3.40	**62.84**±0.64	60.92±1.72
		1	61.41±1.60	**67.13**±2.59 †	66.76±1.25 †
		2	49.79±0.51	67.17±5.26 †	**68.36**±2.70 †
		3	35.67±0.30	41.96±2.95	**42.93**±2.30 †
		4	57.68±0.26	66.01±1.88 †	**67.99**±2.30 †
	Splice Site Prediction	reconstructed	78.64±0.43	80.32±1.24 †	**80.76**±1.06 †
	Mouse Transcription Factor Prediction	0	22.51±2.78	46.09±2.17 †	**47.96**±5.01 †
		1	76.56±0.51	**78.93**±0.31 †	76.68±0.81
		2	62.69±1.52	74.76±3.07 †	**74.78**±2.98 †
		3	36.93±8.35	**68.61**±4.24 †	66.58±3.24 †
		4	30.23±3.13	40.07±0.83 †	**40.57**±2.09 †
	Covid Variant Classification	Covid	**66.43**±0.48 †	36.71±9.69	14.81±0.46
	Core Promoter Detection	tata	78.26±2.85	79.54±1.61	**79.87**±2.50
		notata	**66.60**±1.07	66.52±0.28	65.95±0.51
		all	66.47±0.74	65.26±1.11	**67.16**±0.55
	Promoter Detection	tata	78.58±3.39	**79.74**±2.66	78.77±0.78
		notata	**90.81**±0.51	89.86±0.76	90.28±0.37
		all	**88.00**±0.39	87.60±0.51	87.93±0.76
GB	Demo	coding vs intergenomic seqs	75.14±0.35	80.04±0.28 †	**80.25**±0.24 †
		human or worm	89.89±0.15	92.65±0.11 †	**92.71**±0.27 †
	Enhancers	drosophila enhancers stark	7.99±3.01	10.77±2.34	**10.87**±3.32
		human enhancers cohn	30.76±2.05	46.63±0.88 †	**46.68**±1.11 †
		human enhancers ensembl	**79.48**±0.10 †	44.48±2.94	72.99±0.36
	Regulatory	human ensembl regulatory	89.73±0.21	89.91±0.72	**90.21**±1.37
		human non-tata promoters*	64.98±0.21	**83.57**±0.73 †	79.90±1.48 †
	Open Chromatin Regions	human ocr ensembl	39.92±0.85	**56.22**±0.28 †	55.36±2.52 †

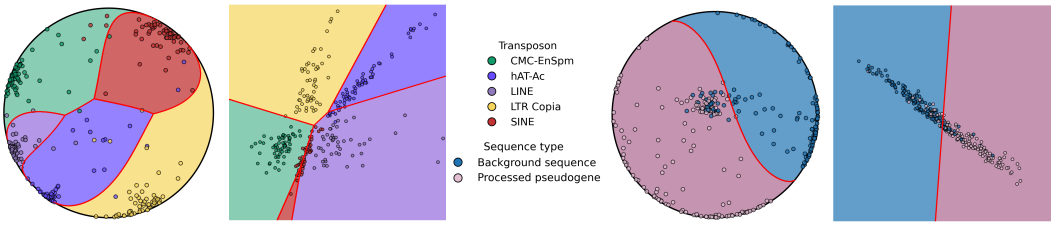

Figure 3: The decision boundaries learned by 2-dimensional HCNNs (circles) and CNNs (squares). Boundaries for transposon sequences and processed pseudogenes are visualized on the Poincaré disk and Euclidean plane. Regions are colored according to their predicted class labels, whereas points are colored with respect to their true class labels.

tendencies. However, we observe that the increase in values of δ_{worst} are only weakly anticorrelated with relative improvements in performance on learning tasks ($r_S = -0.35, r_M = -0.21$, Appendix Figure 10). An outlier to this pattern appears to be the Covid dataset, which has low hyperbolicity and poor performance in HCNNs.

Previous studies have attempted to calibrate their reported δ_{worst} values by comparing them to empirical estimates of δ_{worst} for the Poincaré disk $\mathbb{D}^2$, and the 2-sphere S^2 (Khrulkov et al., 2020; Yang et al., 2024), however we note that these empirical estimates are for metric spaces that are categorically much lower in dimensionality than the feature spaces used for the dataset embeddings, leading to potentially incongruous comparisons.

Table 2: Model performance (MCC) under different synthetic data-generating scenarios (with the same notation as Table 1).

Scenario	Sequence	Model: Euclidean CNN	Model: Hyperbolic HCNN-S	Model: Hyperbolic HCNN-M
A	Artificial	62.38±2.28	**65.25**±3.27	59.25±2.60
	Real	61.72±3.08	**66.44**±3.14	61.26±2.99
B	Artificial	58.50±0.82	**60.53**±0.80	59.75±0.54
	Real	57.50±0.88	**62.53**±6.94	59.12±0.54
C	Artificial	62.05±1.62	**67.65**±1.09 †	67.43±1.57 †
	Real	66.22±0.44	**73.62**±0.62 †	69.30±2.34 †

Indeed, we find evidence that high-dimensional data may lead to "emergent hyperbolicity," with points at higher dimensions producing smaller δ_{worst} and δ_{avg} values (Appendix Figures 12 and 13). Our results highlight a pronounced disparity: the difference in empirical δ values between embeddings sampled on $\mathbb{H}^2$ and those sampled on higher-dimensional hyperbolic spaces ($\mathbb{H}^d$, where $d \in [200, 1000]$) – with comparable magnitudes to the sequence embeddings – can be as large as 0.2 (Appendix Figure 12). This disparity becomes even more pronounced on Euclidean ($\mathbb{R}^d$) and hyperspherical ($\mathbb{S}^d$) manifolds. Such significant differences in δ values may largely determine whether the estimated δ indicates a more hyperbolic nature of the underlying space or otherwise.

To provide a more equitable calibration of hyperbolicity, we compare the δ distributions from our genomic datasets to those from simulated datasets of matching dimensionality. We generate these simulated datasets on both Euclidean ($K = 0$) and hyperbolic ($K = -1$) manifolds. Figure 11 illustrates the δ distributions for each set of dataset embeddings, where each embedding $G \in \mathbb{R}^{528}$. Our results reveal that the majority of the genomic dataset embeddings exhibit greater hyperbolicity (lower δ values) compared to embeddings simulated from a baseline Gaussian distribution on a Euclidean manifold of the same dimensionality. To quantify this difference, we employ the Wilcoxon rank-sum test between the baseline and the genome dataset distributions. This analysis shows that 25 out of 43 sequence datasets have significantly lower δ values than the baseline ($p < 0.05$). These findings lend credence to the hypothesis that genomic sequence data may possess an innate hyperbolicity, making them better suited to hyperbolic representations.

Our approach of examining the entire distribution of δ values, rather than relying on a single scalar measure, reveals nuanced insights into the hyperbolic tendencies of different datasets. This comprehensive view allows us to capture subtleties that might otherwise be overlooked. For instance,

the H3K36me3 dataset exhibits a δ distribution that is significantly lower in hyperbolicity compared to the baseline. However, its high δ_{worst} estimate suggests that it may be less hyperbolic than the baseline when considering only this single metric. Similarly, while the TEB datasets show relatively large δ_{worst} estimates, their δ distributions are notably right-skewed. These characteristics appear more consistent with the superior performance of HCNN models on these datasets.

The discrepancies between single-point estimates (δ_{worst}, δ_{avg}) and the full distributions underscore the importance of a more holistic approach. By considering the entire spectrum of δ values across the feature space, we gain a more accurate characterization of the data's tree-like properties. This comprehensive perspective not only provides a richer understanding of the dataset's geometric structure but also offers better insights into why certain models, such as HCNNs, perform well on these datasets. Finally, in expanding our analysis to DNA LMs in section A.9.3, we observe that these characteristics extend to a wide range of models.

6 Conclusion

We present a novel application of hyperbolic CNNs for genomic sequence modeling, thoroughly examining both the strengths and limitations of this approach. Our findings demonstrate that hyperbolic embeddings provide a distinct performance advantage in key genomics tasks, particularly when working within resource constraints. Additionally, our investigation into the hyperbolicity of dataset embeddings reveals meaningful correlations between dimensionality and δ-hyperbolicity, further underscoring the utility of hyperbolic space for genome representation.

While our model is relatively simple, this paper lays the groundwork for more sophisticated models that could further harness the strengths of hyperbolic embeddings. CNNs are workhorses of machine learning in genomics, thus using HCNNs in more specialized genomics challenges could further improve performance.

Moreover, this paper sets the stage for future research aimed at developing more robust metrics for quantifying and assessing the hyperbolicity of dataset embeddings. We have only begun to explore the relationship between hyperbolicity, curvature, and dimensionality, and these properties would greatly benefit from formalization and rigorous testing. Our work opens up new avenues for understanding and optimizing hyperbolic models in genomics, encouraging further exploration into this promising paradigm.

References

Réka Albert, Bhaskar DasGupta, and Nasim Mobasheri. Topological implications of negative curvature for biological and social networks. *Physical Review E*, 89(3):032811, 2014.

Žiga Avsec, Vikram Agarwal, Daniel Visentin, Joseph R Ledsam, Agnieszka Grabska-Barwinska, Kyle R Taylor, Yannis Assael, John Jumper, Pushmeet Kohli, and David R Kelley. Effective gene expression prediction from sequence by integrating long-range interactions. *Nature methods*, 18 (10):1196–1203, 2021.

Ahmad Bdeir, Kristian Schwethelm, and Niels Landwehr. Fully hyperbolic convolutional neural networks for computer vision. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=ekz1hN5QNh.

Gary Bécigneul and Octavian-Eugen Ganea. Riemannian adaptive optimization methods. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=r1eiqi09K7.

Michele Borassi, Alessandro Chessa, and Guido Caldarelli. Hyperbolicity measures democracy in real-world networks. *Physical Review E*, 92(3):032812, 2015.

Nathan J Bowen and I King Jordan. Transposable elements and the evolution of eukaryotic complexity. *Current issues in molecular biology*, 4(3):65–76, 2002.

Marta Byrska-Bishop, Uday S Evani, Xuefang Zhao, Anna O Basile, Haley J Abel, Allison A Regier, André Corvelo, Wayne E Clarke, Rajeeva Musunuri, Kshithija Nagulapalli, et al. High-coverage whole-genome sequencing of the expanded 1000 genomes project cohort including 602 trios. *Cell*, 185(18):3426–3440, 2022.

Benjamin Paul Chamberlain, James R. Clough, and Marc Peter Deisenroth. Neural embeddings of graphs in hyperbolic space. *CoRR*, abs/1705.10359, 2017. URL http://arxiv.org/abs/1705.10359.

Ines Chami, Zhitao Ying, Christopher Ré, and Jure Leskovec. Hyperbolic graph convolutional neural networks. *Advances in neural information processing systems*, 32, 2019.

Ines Chami, Albert Gu, Vaggos Chatziafratis, and Christopher Ré. From trees to continuous embeddings and back: Hyperbolic hierarchical clustering. *Advances in Neural Information Processing Systems*, 33:15065–15076, 2020a.

Ines Chami, Adva Wolf, Da-Cheng Juan, Frederic Sala, Sujith Ravi, and Christopher Ré. Low-dimensional hyperbolic knowledge graph embeddings. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pp. 6901–6914. Association for Computational Linguistics, 2020b. doi:10.18653/V1/2020.ACL-MAIN.617. URL https://doi.org/10.18653/v1/2020.acl-main.617.

Sourav Chatterjee and Leila Sloman. Average gromov hyperbolicity and the parisi ansatz. *Advances in Mathematics*, 376:107417, 2021.

Kathleen M Chen, Aaron K Wong, Olga G Troyanskaya, and Jian Zhou. A sequence-based global map of regulatory activity for deciphering human genetics. *Nature genetics*, 54(7):940–949, 2022a.

Weize Chen, Xu Han, Yankai Lin, Hexu Zhao, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. Fully hyperbolic neural networks. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pp. 5672–5686. Association for Computational Linguistics, 2022b. doi:10.18653/V1/2022.ACL-LONG.389. URL https://doi.org/10.18653/v1/2022.acl-long.389.

Weize Chen, Xu Han, Yankai Lin, Kaichen He, Ruobing Xie, Jie Zhou, Zhiyuan Liu, and Maosong Sun. Hyperbolic pre-trained language model. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2024.

Philippe Chlenski, Quentin Chu, Raiyan R Khan, Antonio Khalil Moretti, and Itsik Pe'er. Mixed-curvature decision trees and random forests. *arXiv preprint arXiv:2410.13879*, 2024.

Nathann Cohen, David Coudert, and Aurélien Lancin. On computing the gromov hyperbolicity. *Journal of Experimental Algorithmics (JEA)*, 20:1–18, 2015.

Micaela E Consens, Cameron Dufault, Michael Wainberg, Duncan Forster, Mehran Karimzadeh, Hani Goodarzi, Fabian J Theis, Alan Moses, and Bo Wang. To transformers and beyond: large language models for the genome. *arXiv preprint arXiv:2311.07621*, 2023.

Gregory M Cooper, Eric A Stone, George Asimenos, Eric D Green, Serafim Batzoglou, and Arend Sidow. Distribution and intensity of constraint in mammalian genomic sequence. *Genome research*, 15(7):901–913, 2005.

Gabriele Corso, Zhitao Ying, Michal Pándy, Petar Veličković, Jure Leskovec, and Pietro Liò. Neural distance embeddings for biological sequences. *Advances in Neural Information Processing Systems*, 34:18539–18551, 2021.

René Dreos, Giovanna Ambrosini, Rouayda Cavin Périer, and Philipp Bucher. Epd and epdnew, high-quality promoter resources in the next-generation sequencing era. *Nucleic acids research*, 41 (D1):D157–D164, 2013.

Denise Drongitis, Francesco Aniello, Laura Fucci, and Aldo Donizetti. Roles of transposable elements in the different layers of gene expression regulation. *International Journal of Molecular Sciences*, 20(22):5755, 2019.

Kseniia Dudnyk, Donghong Cai, Chenlai Shi, Jian Xu, and Jian Zhou. Sequence basis of transcription initiation in the human genome. *Science*, 384(6694):eadj0116, 2024.

Geoffrey J Faulkner, Yasumasa Kimura, Carsten O Daub, Shivangi Wani, Charles Plessy, Katharine M Irvine, Kate Schroder, Nicole Cloonan, Anita L Steptoe, Timo Lassmann, et al. The regulated retrotransposon transcriptome of mammalian cells. *Nature Genetics*, 41(5):563–571, 2009. doi:10.1038/ng.368.

Hervé Fournier, Anas Ismail, and Antoine Vigneron. Computing the gromov hyperbolicity of a discrete metric space. *Information Processing Letters*, 115(6-8):576–579, 2015.

Adam Frankish, Mark Diekhans, Anne-Maud Ferreira, Rory Johnson, Irwin Jungreis, Jane Loveland, Jonathan M Mudge, Cristina Sisu, James Wright, Joel Armstrong, et al. Gencode reference annotation for the human and mouse genomes. *Nucleic acids research*, 47(D1):D766–D773, 2019.

Octavian Ganea, Gary Bécigneul, and Thomas Hofmann. Hyperbolic neural networks. *Advances in neural information processing systems*, 31, 2018.

Katarína Grešová, Vlastimil Martinek, David Čechák, Petr Šimeček, and Panagiotis Alexiou. Genomic benchmarks: a collection of datasets for genomic sequence classification. *BMC Genomic Data*, 24(1):25, 2023.

M Gromov. Hyperbolic groups. *Essays in Group Theory, pages/Springer-Verlag*, 1987.

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

Aric Hagberg, Pieter J Swart, and Daniel A Schult. Exploring network structure, dynamics, and function using networkx. Technical report, Los Alamos National Laboratory (LANL), Los Alamos, NM (United States), 2008.

Dustin C Hancks and Haig H Kazazian. Mobilization of transposable elements by environmental and endogenous factors. *Human Molecular Genetics*, 25(R2):R45–R50, 2016. doi:10.1093/hmg/ddw025.

Alexander Hayward and Clément Gilbert. Transposable elements. *Current Biology*, 32(17):R904–R909, 2022.

Joy Hsu, Jeffrey Gu, Gong Wu, Wah Chiu, and Serena Yeung. Capturing implicit hierarchical structure in 3d biomedical images with self-supervised hyperbolic representations. *Advances in neural information processing systems*, 34:5112–5123, 2021.

Jaime Huerta-Cepas, François Serra, and Peer Bork. Ete 3: reconstruction, analysis, and visualization of phylogenomic data. *Molecular biology and evolution*, 33(6):1635–1638, 2016.

Timothy Hughes, Young Hyun, and David A Liberles. Visualising very large phylogenetic trees in three dimensional hyperbolic space. *BMC bioinformatics*, 5:1–6, 2004.

Yanrong Ji, Zhihan Zhou, Han Liu, and Ramana V Davuluri. Dnabert: pre-trained bidirectional encoder representations from transformers model for dna-language in genome. *Bioinformatics*, 37 (15):2112–2120, 2021.

Yueyu Jiang, Puoya Tabaghi, and Siavash Mirarab. Learning hyperbolic embedding for phylogenetic tree placement and updates. *Biology*, 11(9):1256, 2022a.

Yueyu Jiang, Puoya Tabaghi, and Siavash Mirarab. Phylogenetic placement problem: A hyperbolic embedding approach. In *RECOMB International Workshop on Comparative Genomics*, pp. 68–85. Springer, 2022b.

Martin E J"onsson, Rebecca Garza, Per A Johansson, and Johan Jakobsson. Transposable elements: a common feature of neurodegenerative disorders. *Mobile DNA*, 11(1):1–15, 2020. doi:10.1186/s13100-020-00207-x.

Valentin Khrulkov, Leyla Mirvakhabova, Evgeniya Ustinova, Ivan Oseledets, and Victor Lempitsky. Hyperbolic image embeddings. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6418–6428, 2020.

Reinmar J Kobler, Jun-ichiro Hirayama, and Motoaki Kawanabe. Controlling the fréchet variance improves batch normalization on the symmetric positive definite manifold. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3863–3867. IEEE, 2022.

Guy Lebanon and John Lafferty. Hyperplane margin classifiers on the multinomial manifold. In *Proceedings of the twenty-first international conference on Machine learning*, pp. 66, 2004.

Shaoteng Liu, Jingjing Chen, Liangming Pan, Chong-Wah Ngo, Tat-Seng Chua, and Yu-Gang Jiang. Hyperbolic visual embedding learning for zero-shot recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9273–9281, 2020.

Marie Lopez, Hugo Dalla-Torre, Liam Gonzalez, Javier Mendoza-Revilla, Nicolas Lopez Carranza, Adam Grzywaczewski, Francesco Oteri, Christian Dallago, Evan Trop, Hassan Sirelkhatim, et al. The nucleotide transformer: Building and evaluating robust foundation models for human genomics. 2023.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL `https://openreview.net/forum?id=Bkg6RiCqY7`.

Aaron Lou, Isay Katsman, Qingxuan Jiang, Serge Belongie, Ser-Nam Lim, and Christopher De Sa. Differentiating through the fréchet mean. In *International conference on machine learning*, pp. 6393–6403. PMLR, 2020.

Amy X Lu, Alex X Lu, and Alan Moses. Evolution is all you need: phylogenetic augmentation for contrastive learning. *arXiv preprint arXiv:2012.13475*, 2020.

Xizhi Luo, Shiyu Chen, and Yu Zhang. Plantrep: a database of plant repetitive elements. *Plant cell reports*, pp. 1–4, 2022.

Emile Mathieu, Charline Le Lan, Chris J Maddison, Ryota Tomioka, and Yee Whye Teh. Continuous hierarchical representations with poincaré variational auto-encoders. *Advances in neural information processing systems*, 32, 2019.

Yoshihiro Nagano, Shoichiro Yamaguchi, Yasuhiro Fujita, and Masanori Koyama. A wrapped normal distribution on hyperbolic space for gradient-based learning. In *International Conference on Machine Learning*, pp. 4693–4702. PMLR, 2019.

Eric Nguyen, Michael Poli, Marjan Faizi, Armin Thomas, Michael Wornow, Callum Birch-Sykes, Stefano Massaroli, Aman Patel, Clayton Rabideau, Yoshua Bengio, et al. Hyenadna: Long-range genomic sequence modeling at single nucleotide resolution. *Advances in neural information processing systems*, 36, 2024.

Maximillian Nickel and Douwe Kiela. Learning continuous hierarchies in the lorentz model of hyperbolic geometry. In *International conference on machine learning*, pp. 3779–3788. PMLR, 2018.

Eric Qu and Dongmian Zou. Autoencoding hyperbolic representation for adversarial generation. *arXiv preprint arXiv:2201.12825*, 2022.

Lukas Schrader and Jürgen Schmitz. The impact of transposable elements in adaptive evolution. *Molecular Ecology*, 28(6):1537–1549, 2019.

Shihao Shen, Lan Lin, James J Cai, Peng Jiang, Emily J Kenkel, Miranda R Stroik, Shigeo Sato, Beverly L Davidson, and Yi Xing. Widespread establishment and regulatory impact of alu exons in human genes. *Proceedings of the National Academy of Sciences*, 108(7):2837–2842, 2011. doi:10.1073/pnas.1012834108.

Ondrej Skopek, Octavian-Eugen Ganea, and Gary Bécigneul. Mixed-curvature variational autoencoders. In *8th international conference on learning representations (ICLR 2020)(virtual)*. International Conference on Learning Representations, 2020.

Stephanie J Spielman and Claus O Wilke. Pyvolve: a flexible python module for simulating sequences along phylogenies. *PloS one*, 10(9):e0139047, 2015.

Vasavi Sundaram, Yong Cheng, Zhihai Ma, Daofeng Li, Xiaoyun Xing, Peter Edge, Michael P Snyder, and Ting Wang. Widespread contribution of transposable elements to the innovation of gene regulatory networks. *Genome Research*, 24(12):1963–1976, 2014. doi:10.1101/gr.168872.113.

Simon Tavaré. Line-of-descent and genealogical processes, and their applications in population genetics models. *Theoretical population biology*, 26(2):119–164, 1984.

Felix Teufel, Magnús Halldór Gíslason, José Juan Almagro Armenteros, Alexander Rosenberg Johansen, Ole Winther, and Henrik Nielsen. Graphpart: homology partitioning for biological sequence analysis. *NAR genomics and bioinformatics*, 5(4):lqad088, 2023.

Tian Tian, Cheng Zhong, Xiang Lin, Zhi Wei, and Hakon Hakonarson. Complex hierarchical structures in single-cell genomics data unveiled by deep hyperbolic manifold learning. *Genome Research*, 33(2):232–246, 2023.

Alexandru Tifrea, Gary Bécigneul, and Octavian-Eugen Ganea. Poincare glove: Hyperbolic word embeddings. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL `https://openreview.net/forum?id=Ske5r3AqK7`.

Jonathan N Wells and Cédric Feschotte. A field guide to eukaryotic transposable elements. *Annual review of genetics*, 54(1):539–561, 2020.

Menglin Yang, Aosong Feng, Bo Xiong, Jiahong Liu, Irwin King, and Rex Ying. Enhancing llm complex reasoning capability through hyperbolic geometry. In *ICML 2024 Workshop on LLMs and Cognition*, 2024.

Tianwei Yue, Yuanxin Wang, Longxiang Zhang, Chunming Gu, Haoru Xue, Wenping Wang, Qi Lyu, and Yujie Dun. Deep learning for genomics: From early neural nets to modern large language models. *International Journal of Molecular Sciences*, 24(21):15858, 2023.

Jian Zhou. Sequence-based modeling of three-dimensional genome architecture from kilobase to chromosome scale. *Nature genetics*, 54(5):725–734, 2022.

Yuansheng Zhou and Tatyana O Sharpee. Hyperbolic geometry of gene expression. *Iscience*, 24(3), 2021.

Zhihan Zhou, Yanrong Ji, Weijian Li, Pratik Dutta, Ramana V. Davuluri, and Han Liu. DNABERT-2: efficient foundation model and benchmark for multi-species genomes. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL `https://openreview.net/forum?id=oMLQB4EZE1`.

A APPENDIX

A.1 LORENTZ CONVOLUTIONAL LAYER

A.1.1 LAYER COMPONENTS

We further break down the Lorentz convolutional layer by defining each separate transformation. First, given hyperbolic points $\{\mathbf{x}_i\}_{i=1}^N$, the Lorentz direct concatenation (HCat) (Qu & Zou, 2022) is given by:

$$\mathbf{y} = \text{HCat}(\{\mathbf{x}_i\}_{i=1}^N) = \left[\sqrt{\sum_{i=1}^{N} x_{i_t}^2 + \frac{N-1}{K}}, \mathbf{x}_{1_s}^T, \ldots, \mathbf{x}_{N_s}^T \right],^T \tag{10}$$

with $\mathbf{y} \in \mathbb{L}_K^{nN} \subset \mathbb{R}^{nN+1}$. This manipulation represents a numerically stable way to concatenate hyperbolic representations. Next, Chen et al. (2022b) derived a Lorentz fully-connected layer where given the input vector $\mathbf{x}$ and the weight parameters $\mathbf{W} \in \mathbb{R}^{m \times n+1}, \boldsymbol{v} \in \mathbb{R}^{n+1}$ for the fully connected layer, the transformation matrix is defined as:

$$f_x \left(\left[\begin{array}{c} \boldsymbol{v}^T \\ \mathbf{W} \end{array} \right] \right) = \left[\begin{array}{c} \frac{\sqrt{\|\mathbf{W}x\|^2 - 1/K}}{v^T x} \boldsymbol{v}^T \\ \mathbf{W} \end{array} \right] \tag{11}$$

Then, incorporating normalization gives

$$\mathbf{y} = \text{LFC}(x) = \left[\frac{\sqrt{\|\psi(\mathbf{W}\mathbf{x} + \mathbf{b})\|^2 - 1/K}}{\psi(\mathbf{W}\mathbf{x} + \mathbf{b})} \right] \tag{12}$$

with operation function

$$\phi(\mathbf{W}\mathbf{x}, v) = \lambda \sigma(v^T x + b') \frac{\mathbf{W}\psi(x) + b}{\|\mathbf{W}\psi(x) + b\|} \tag{13}$$

where $\lambda > 0$ is a learnable scaling parameter and $b \in \mathbb{R}^n$, ψ, σ denote the bias, activation, and sigmoid function, respectively.

A.1.2 LAYER MAPPING

HCNN-M models leverage multiple manifolds $[K_1, ..., K_u]$ for each of u designated blocks. Therefore, we define the mapping between manifolds as follows, using the definitions of exponential and logarithmic maps defined in equations 1 and 2, respectively. For a mapping of point $\mathbf{x} \in \mathcal{M}_1$, where manifold $\mathcal{M}_1$ has curvature K_1, to manifold $\mathcal{M}_2$ with curvature K_2, we must first apply a logarithmic map to bring $\mathbf{x}$ to the tangent space $T_{\mathbf{0}}\mathcal{M}_1$ at the origin. Then, we perform an exponential mapping of the resulting point from the tangent space at the origin to the new manifold $\mathcal{M}_2$. The layer map operation $\mathbf{LM}_{\mathcal{M}_1 \to \mathcal{M}_2}(\mathbf{x})$ can therefore be defined as follows:

$$\mathbf{LM}_{\mathcal{M}_1 \to \mathcal{M}_2}(\mathbf{x}) = \exp_{\mathbf{0}}^{K_2}(\log_{\mathbf{0}}^{K_1}(\mathbf{x})) \tag{14}$$

A.2 MODELING DETAILS

A.2.1 MODEL

A detailed breakdown of the CNN/HCNN model architecture is visualized in Figure 4. The HCNNs use the Lorentz formulation of each model component. For HCNN-M, we show the partition of each manifold across each segment of the architecture. We use cross-entropy loss for our objective, and train each model end-to-end on each dataset.

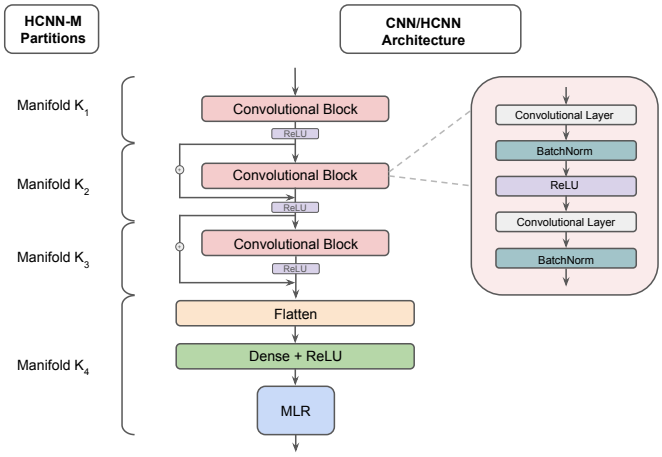

Figure 4: The generalized block architecture for the CNNs/HCNNs. On the left, we delineate the manifold partitions used for our HCNN-M models.

A.2.2 Hyperparameters

When possible, we keep the hyperparameters constant across the different model types (Table 3). However, we train the Euclidean CNN with the AdamW optimizer (Loshchilov & Hutter, 2019) and the HCNNs with RiemannianAdam (Bécigneul & Ganea, 2019).

Table 3: Hyperparameter settings for CNN/HCNN training.

	Euclidean CNN	HCNN-S	HCNN-M
Optimizer	AdamW	RiemannianAdam	RiemannianAdam
Learning Rate (TEB/GUE/GB)	1e-4, 1e-4, 1e-5	1e-4, 1e-4, 1e-5	1e-4, 1e-4, 1e-5
Manifold Learning Rate	N/A	2e-2	2e-2
Batch size	100	100	100
Weight decay	0.1	0.1	0.1
Epochs	100	100	100
β_1, β_2	0.9, 0.999	0.9, 0.999	0.9, 0.999

A.3 Transposable Elements Benchmark

TEB presents seven distinct sequence classification datasets categorized within three prediction tasks. An overview of the datasets are presented in Table 4. Sequence and annotation data were integrated from both human and plant genome datasets. TEB is publicly available online.

For the retrotransposon and DNA transposon tasks, we craft a dataset by employing annotations from PlantRep (Luo et al., 2022), a database that provides comprehensive annotations of plant repetitive elements across 459 plant genomes. We narrowed the number of candidate species to those that had an appropriate number of TEs of interest to power deep learning tasks, as well as an average TE sequence length of similar magnitude to the other benchmark datasets (200-1000 bp). Then, we randomly selected *Oryza glumipatula* from the set of candidate species to use as as the plant species for our benchmark. Annotations were downloaded from PlantRep, while the *Oryza glumipatula* genome (v1.5) was downloaded from the NCBI genome browser (`https://ftp.ncbi.nlm.nih.gov`). Within the retrotransposon group, we study LTR Copia, LINEs, and SINEs. LTR Copia are a type of retrotransposon characterized by a pair of identical flanking repetitive regions called long terminal repeats (LTRs). Conversely, long interspersed nuclear elements (LINEs), and short interspersed nuclear elements (SINEs) are retrotransposons that do not contain LTRs, and generally contain a promoter while varying by length. Next, within the DNA transposon group, we target two of the most

ubiquitous sub-families: CMC-EnSpm and hAT-Ac, each of which are distinguished by specific short terminal inverted repeats.

While pseudogenes themselves are not a type of TE, they are often the result of TE activity. Therefore, we examine the presence of pseudogenes in the human reference genome (GRCh38.p12), using gene/transcript biotype annotations from GENCODE and Ensembl (Frankish et al., 2019). Pseudogenes are classified as processed and unprocessed, each of which are the result of a different mechanism of action. A processed pseudogene lacks introns and arises from reverse transcription of mRNA and then reinsertion of DNA into the genome, while an unprocessed pseudogene may contain introns and is the product of a gene duplication event.

For dataset construction, we created a positive set of sequences spanning each TE of interest. We then generated a negative set by randomly sampling non-overlapping, remaining portions of the genome (without replacement) until we had a matching number of negative sequences. We used a chromosome level train/validation/test split for our sequences, separating out chromosomes 8/9 and 20-22/17-19 for validation/test in *Oryza glumipatula* and human, respectively, while the remaining chromosomes are used for training.

Table 4: Summary statistics for TEB, including the specific type of TE and the number of training, validation, and test samples in each dataset.

Prediction Task	Species	Max Length	Datasets	Train / Dev / Test
Retrotransposons	Plant	500	LTR Copia	7666 / 682 / 568
		1000	LINEs	22502 / 2030 / 1782
		500	SINEs	21152 / 1836 / 1784
DNA Transposons	Plant	200	CMC-EnSpm	19912 / 1872 / 1808
		1000	hAT-Ac	17322 / 1822 / 1428
Pseudogenes	Human	1000	processed	17956 / 1046 / 1740
		1000	unprocessed	12938 / 766 / 884

A.4 DNA Language Models

We compare the classification performance of our HCNN models to the performance of several DNA LMs, as reported in (Zhou et al., 2024). Table 5 documents the performance of eight large DNA LMs on a subset of GUE datasets, as well as the number of trainable parameters present in each model. We provide a short description of each model:

DNABERT (5-mer, 6-mer): An early iteration of a pretrained transformer model for the genome, DNABERT (Ji et al., 2021) uses the BERT architecture and is trained on human DNA sequences. There are four variants of the model, and here we list the results for the 5-mer and 6-mer versions, which use overlapping 5/6-mer tokenization of sequences.

NT (500M human, 500M 1000g, 2500M 1000g, 2500M multi): NT represents the largest class of models in terms of parameters and training data. There are four variants of NT. The labels "500M" and "2500M" refer to the number of trainable parameters in the model. For the training data, the categories "human", "1000g", and "multi" refer to the human reference genome, the 3203 human genomes from the 1000 Genome project (Byrska-Bishop et al., 2022), and genomes from 850 different species, respectively.

DNABERT-2, DNABERT-2-PT: A refinement over DNABERT, DNABERT-2 incorporates Byte-Pair Encoding and several architectural upgrades for improved learning capabilities. DNABERT-2 is pretrained on the human reference genome, while DNABERT-2-PT is further pretrained on the training sets of the 28 GUE datasets.

A.5 Manifold Curvature

Figure 6 depicts the learned curvatures for models trained on TEB. In the HCNN-M models, blocks 1-3 represent each hyperbolic convolutional block in the model, which have a corresponding manifold with its own curvature. Block 4 represents the portion of the model that involves flattening, a dense

Table 5: The performance (MCC) of several prominent DNA LMs in comparison to the HCNNs on GUE. The best performing score for each GUE dataset is bolded.

	Caduceus -Ph	Hyena DNA	DNA BERT (5-mer)	DNA BERT (6-mer)	NT -500M human	NT -500M 1000g	NT -2500M 1000g	NT -2500M multi	DNA BERT-2	DNA BERT-2 -PT	HCNN -S	HCNN -M
Parameters	7.7M	28.2M	87M	89M	500M	500M	2.5B	2.5B	117M	117M	6.6M	6.6M
H3	77.09	67.17	73.40	73.10	69.67	72.52	74.61	78.77	78.27	**80.17**	69.42	69.95
H3K14ac	41.44	31.98	40.68	40.06	33.55	39.37	44.08	**56.20**	52.57	57.42	56.03	48.25
H3K36me3	46.49	48.27	48.29	47.25	44.14	45.58	50.86	**61.99**	56.88	61.90	55.27	45.76
H3K4me1	37.76	35.83	40.65	41.44	37.15	40.45	43.10	**55.30**	50.52	53.00	41.86	39.78
H3K4me2	28.16	25.81	30.67	32.27	30.87	31.05	30.28	36.49	31.13	39.89	**43.88**	31.27
H3K4me3	24.40	23.15	27.10	27.81	24.06	26.16	30.87	40.34	36.27	41.20	**50.58**	33.59
H3K79me3	60.31	54.09	59.61	61.17	58.35	59.33	61.20	64.70	**67.39**	65.46	64.62	63.35
H3K9ac	52.70	50.84	51.11	51.22	45.81	49.29	52.36	56.01	55.63	**57.07**	54.09	52.25
H4	79.91	73.69	77.27	79.26	76.17	76.29	79.76	81.67	80.71	**81.86**	77.24	76.94
H4ac	40.90	38.44	37.48	37.43	33.74	36.79	41.46	49.13	50.43	50.35	**52.94**	51.86
prom all	85.87	47.38	90.16	90.48	87.71	89.76	90.95	**91.01**	86.77	88.31	88.23	88.83
prom notata	93.23	52.24	92.45	93.05	90.75	91.75	93.07	94.00	94.27	**94.34**	90.92	90.74
prom tata	66.07	5.34	69.51	61.56	78.07	78.23	75.80	79.43	71.59	68.79	**82.70**	79.80
Human TF 0	67.32	62.30	66.97	66.84	61.59	63.64	66.31	66.64	**71.99**	69.12	63.56	63.35
Human TF 1	72.10	67.86	69.98	70.14	66.75	70.17	68.30	70.28	**76.06**	71.87	69.39	68.48
Human TF 2	58.92	46.85	59.03	61.03	53.58	52.73	58.70	58.72	66.52	62.96	**73.80**	71.40
Human TF 3	54.85	41.78	52.95	51.89	42.95	45.24	49.08	51.65	**58.54**	55.35	44.08	43.66
Human TF 4	69.45	61.23	69.26	70.97	60.81	62.82	67.59	69.34	**77.43**	74.94	68.43	70.01
c. prom all	67.28	36.95	69.48	68.90	63.45	66.70	67.39	**70.33**	69.37	67.50	66.33	67.84
c. prom notata	66.07	35.38	69.81	70.47	64.82	67.17	67.46	**71.58**	68.04	69.53	66.78	66.48
c. prom tata	72.94	72.87	76.79	76.06	71.34	73.52	69.66	72.97	74.17	76.18	81.34	**82.07**
Mouse TF 0	56.18	35.62	42.45	44.42	31.04	39.26	48.31	63.31	56.76	**64.23**	48.41	52.31
Mouse TF 1	80.31	80.50	79.32	78.94	75.04	75.49	80.02	83.76	84.77	**86.28**	79.26	77.41
Mouse TF 2	75.89	65.34	62.22	71.44	61.67	64.70	70.14	71.52	79.32	**81.28**	77.86	77.51
Mouse TF 3	73.47	54.20	49.92	44.89	29.17	33.07	42.25	69.44	66.47	73.49	**73.51**	69.73
Mouse TF 4	47.98	19.17	40.34	42.48	29.27	34.01	43.40	47.07	**52.66**	50.80	41.27	43.62
Covid	45.19	23.27	50.46	55.50	50.82	52.06	66.73	**73.04**	71.02	68.49	46.43	16.38
Splice	81.59	72.67	84.02	84.07	79.71	80.97	85.78	**89.35**	84.99	85.93	81.96	82.23

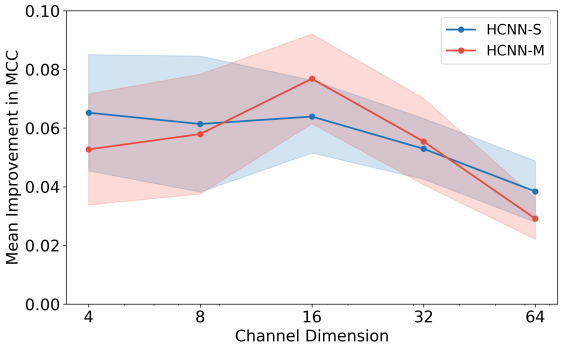

Figure 5: The average improvement in performance (MCC) from HCNNs, as the channel dimension in the convolutional layers varies. HCNN model performance is compared to an equivalent CNN across all TEB classification tasks.

layer, and MLR, operations which all occur on a single hyperbolic manifold (Figure 4). For the HCNN-S models, the value of K is fixed, as a single manifold is used across the entire model.

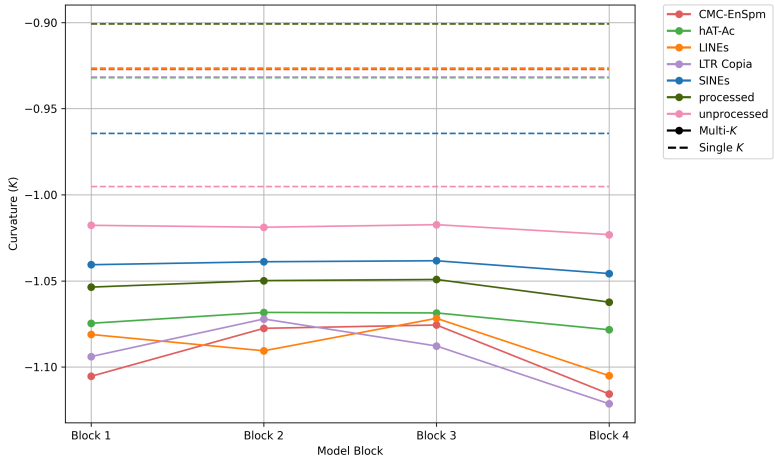

Figure 6: Average values of K, the curvature parameter in the HCNNs, as they vary across each block of the model. These values are reported for models trained on each of the seven classification tasks in TEB.

A.6 Synthetic Datasets

We construct each synthetic dataset by randomly sampling a phylogenetic tree using the ETE (Environment for Tree Exploration) toolkit Huerta-Cepas et al. (2016). To simulate nucleotide sequence evolution along the tree's branches, we use the Pyvolve package (Spielman & Wilke, 2015), specifically for its implementation of the Generalized Time-Reversible (GTR) model (Tavaré, 1984) with default parameters. Four types of fixed-length sequences are generated and used across scenarios A, B, and C:

Artificial tree: The starting ancestral (root) sequence is randomly generated.

Real tree: The starting ancestral sequence is sampled from the human genome.

Artificial background sequence: Sequences are generated randomly and independently by sampling nucleotides.

Real background sequence: Sequences are sampled from independent (different chromosome) regions of the human genome relative to the starting ancestral sequence.

We define the task for each scenario as follows:

(A) **Intra-tree differentiation**: One tree is sampled, with clade membership determining class labels. The model task is to differentiate clades.
(B) **Inter-tree differentiation**: A different tree (with a different starting ancestral sequence) is sampled per label. The model task is to differentiate trees.
(C) **Tree identification**: One tree is sampled, and all sequences from this tree share the same label. Independently sampled background sequences are given a separate label. The model task is to differentiate the tree from the background sequences.

Simulated phylogenetic trees and labels are visualized in Figures 7 and 8. We add noise to the datasets by randomly swapping 10% of the labels in the train and validation sets.

A.7 Homology Splitting

In testing predictive models of biological sequence data, it is common to perform homology splitting where sequences related through their homologous relationships are excluded to determine the model's capacity for generalizability to unseen homology branches. We determine how this partitioning affects HCNNs by assessing the zero-shot capability of our model in identifying sequences originating from an unseen phylogenetic tree against random background sequences.

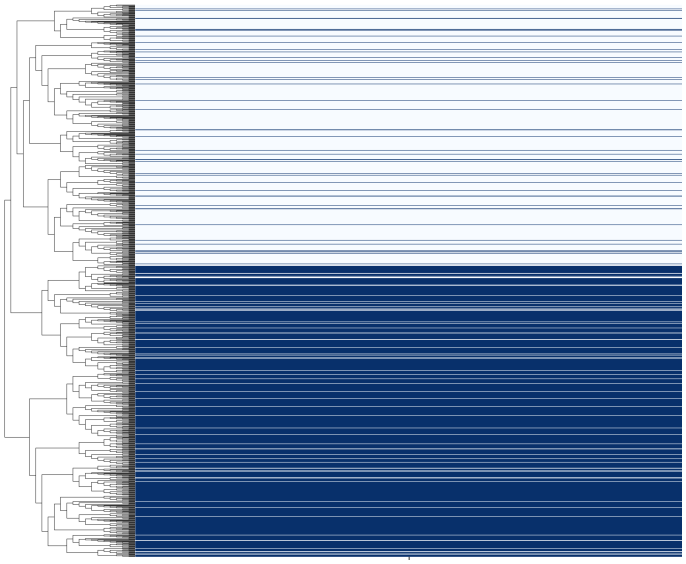

Figure 7: Leaf node sequence classifications (with added noise) in Scenario A for the simulated phylogenetic tree (structure visible left).

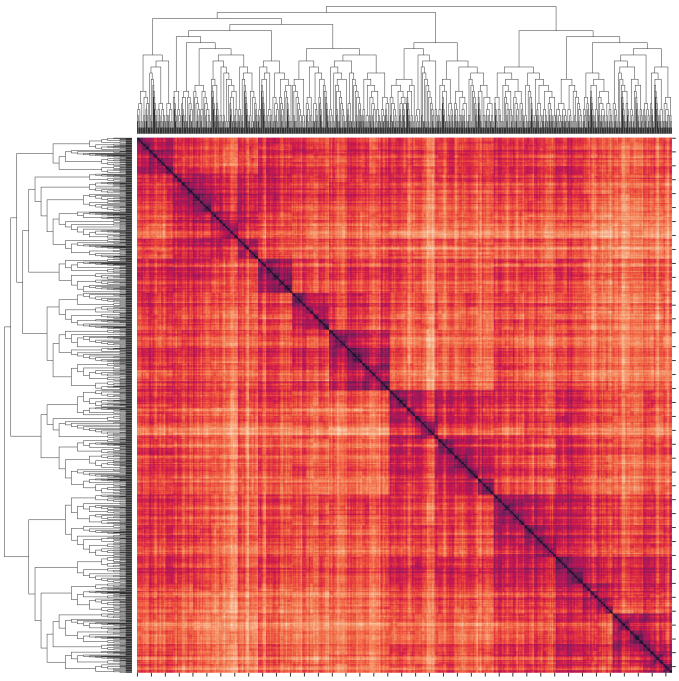

Figure 8: Hamming distance matrix between all leaves in the simulated phylogenetic tree for Scenario A.

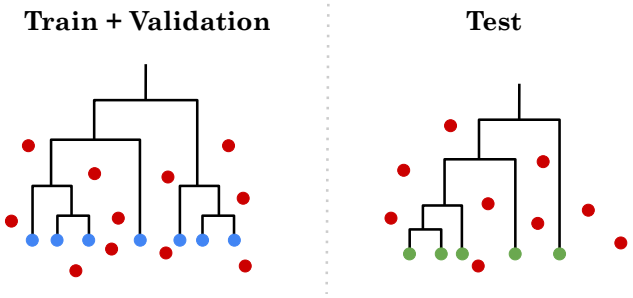

Figure 9: Overview of the homology splitting experiment. A train and validation dataset is generated in the same manner as the scenario C synthetic data. For the test dataset, a completely new tree (and ancestral sequence) is used to generate the tree class.

The experiment setup is visualized in Figure 9. For our training data, we generate a synthetic dataset as we did for testing Scenario C (sequences generated from the tree share the same label, and background sequences not originating from the tree share a different label). However, instead of splitting this one dataset into train/validation/test sequences, we create our test set by generating a completely new phylogenetic tree and sampling sequences from this set. The tree-generated sequences in the test dataset thus originate from completely unseen homology branches.

Results of this experiment are in Table 6. Hyperbolic models gain a significant advantage over the Euclidean model in generalizing to unseen homology branches, which suggests that the inductive biases of a hyperbolic model offer an even larger advantage over Euclidean models than originally estimated, since most genomic datasets do not account for this effect and may therefore overestimate performance of prediction methods (Teufel et al., 2023).

Table 6: Model performance (MCC) on the homology splitting task (with the same notation as Table 1).

CNN	HCNN-S	HCNN-M
24.31±7.99	**45.73**±8.93 †	40.87±8.93 †

A.8 Hybrid Models

Following Bdeir et al. (2024), we experiment with the use of hybrid CNN models, in which we substitute components of our models across manifolds. We construct two hybrid model variants: E2H-CNN and H2E-CNN. In E2H-CNN, we use a Euclidean CNN head and a Lorentzian MLR, whereas H2E-CNN uses a HCNN head and a Euclidean MLR. We compare the performance of the two hybrid models to the other three models in Table 7. On TEB datasets, we observe that the use of a Lorentzian component generally offers an improvement over using a fully Euclidean model, with larger improvements from E2H-CNN. This result would suggest that using hyperbolic hyperplanes to separate classes may be beneficial, even for Euclidean embeddings. Overall, the results show promise in the use of hybrid models.

A.9 δ-Hyperbolicity

A.9.1 Estimation Procedure

Computing δ_{worst} naively is an $\mathcal{O}(n^4)$ operation for a set of n points, therefore we use the efficient approach introduced in Khrulkov et al. (2020) and Cohen et al. (2015). Specifically, we incorporate a sampling procedure to estimate hyperbolicity in an computationally tractable manner. The steps are as follows:

1. Sample N_s points from the dataset (we set $N_s = 1000$).

Table 7: Model performance (MCC) in TEB, averaged over 5 random seeds. The best performing model is bolded.

Dataset	CNN	HCNN-S	HCNN-M	E2H-CNN	H2E-CNN
LTR Copia	54.73±1.45	64.58±3.07	**68.05**±2.80	61.82±2.21	63.95±3.52
LINEs	70.63±1.24	76.12±2.16	77.10±2.92	75.65±0.83	**79.15**±2.36
SINEs	85.15±1.64	85.45±1.16	81.85±2.95	**89.65**±2.13	79.49±3.40
CMC-EnSpm	72.18±0.32	**80.98**±1.48	80.65±1.30	76.75±0.60	77.15±3.43
hAT-Ac	87.45±0.90	89.61±1.34	**91.04**±1.58	89.76±0.85	85.63±1.44
processed	60.66±0.82	**68.30**±0.93	65.41±5.54	66.68±1.31	66.12±0.43
unprocessed	51.94±2.69	56.13±0.56	**58.36**±1.80	58.09±0.96	58.16±1.40

2. Compute the matrix A of pairwise Gromov products using equation 8, and a fixed point $z = z_0$ (detailed in Cohen et al. (2015)).
3. Determine the the matrix $C = (A \otimes A) - A$, where $\otimes$ represents the min-max matrix product: $(A \otimes B)_{ij} = \max \min_k \{A_{ik}, B_{kj}\}$
4. For δ_{worst}, we take the maximum value from C, and for δ_{avg}, we compute the expected value over the unique elements of C pertaining to valid tuples. We apply the scale-invariant transformation mentioned in the main text to the δs in determining the final values reported. However, for the δ_{avg} values, we instead transform the raw values using the scale-invariant ratio introduced in Borassi et al. (2015): $\frac{2\delta_{avg}}{D_{avg}}$, where D_{avg} is the average distance between two randomly selected points.

Results are averaged across multiple runs, and we provide resulting mean and standard deviation. For the genomic datasets, we use the test set of sequence embeddings generated from the final embedding layer of the trained Euclidean CNN models (Table 8).

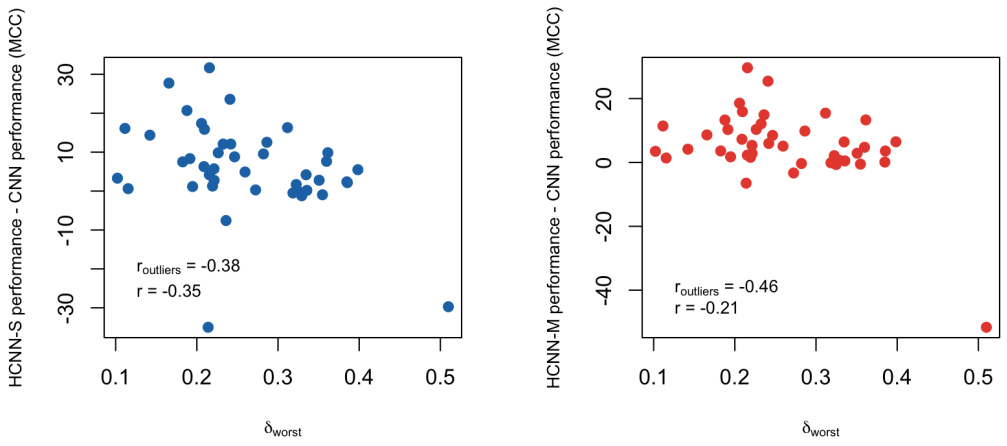

Figure 10: Correlation between δ_{worst} and performance differential between HCCN-S and CNN models. r_{outliers} includes outliers in the Pearson correlation coefficient calculation and r excludes them ($p < 0.05$ except for HCNN-M r).

A.9.2 Metric Space Calibrations

In order to calibrate our δ-hyperbolicity measurements, we scrutinize the behavior of δ approximations at various fixed curvatures (K) and dimensionalities (d). We use the Embedders package, introduced

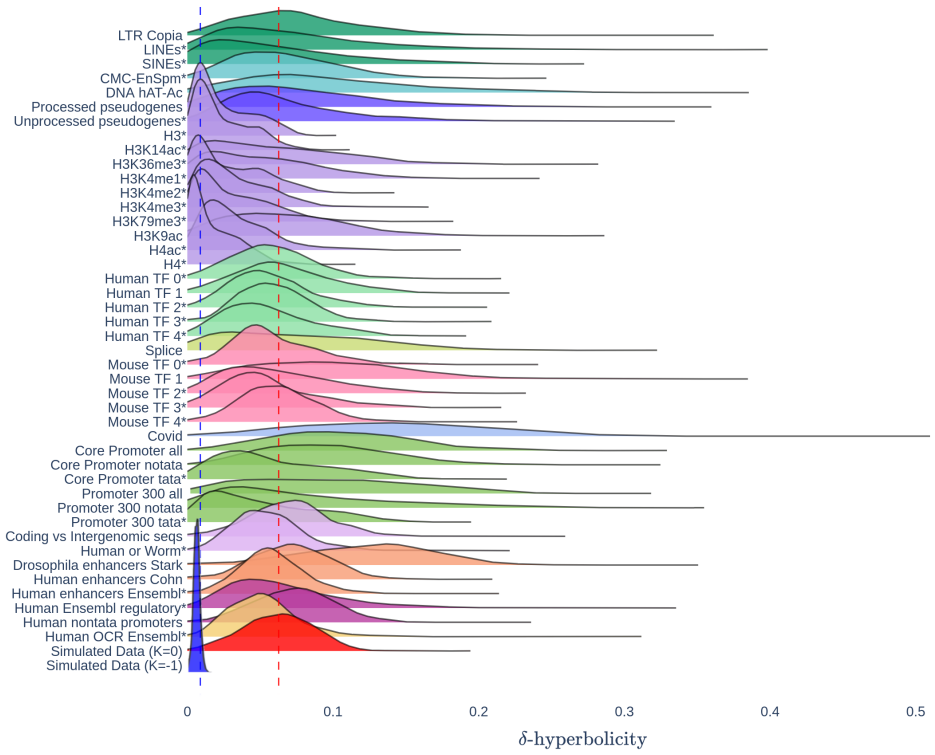

Figure 11: Distribution of scaled δ-hyperbolicity values across each of the genomic datasets. Colors delineate the different task categories, while the bottom two entries provide reference distributions for δs computed from a set of points sampled from the normal distribution on a Euclidean ($K = 0$, red) and hyperbolic ($K = -1$, blue) manifold. Dashed lines indicate the δ_{avg} values for the hyperbolic reference (blue) and the Euclidean reference (red). * Denotes that the corresponding distribution constitutes smaller δ values (is more hyperbolic) than the Euclidean reference based on the Wilcoxon rank-sum test ($p < 0.01$).

in Chlenski et al. (2024), to randomly sample data points from the Gaussian distribution across different manifolds, using the wrapped normal distribution in hyperbolic ($K = -1, -2$) (Nagano et al., 2019) and hyperspherical ($K = 1, 2$) (Skopek et al., 2020) cases. We then compute δ estimates according to the procedure in A.9.1. We use the geodesic distance of each manifold to determine the distance matrix between points.

The results of the simulations are visualized in Figures 12 and 13. The decreasing trend in both δ_{worst} and δ_{avg} estimates (across curvatures) suggests that higher dimensionality of data points may lead to increasing hyperbolicity in datasets. For discrete metric spaces, we confirm that for trees $\delta_{worst} = \delta_{avg} = 0$ by using the NETWORKX package (Hagberg et al., 2008) to generate random tree graphs, and compute the distance matrix based on shortest paths within each graph.

A.9.3 DNA LANGUAGE MODELS

We explore the hyperbolicity of sequences embedded by large DNA LMs. Our analysis encompasses a diverse range of pretrained models, selected to represent various architectural approaches and scales. The models under examination include:

- HyenaDNA: A long-context model that employs a subquadratic alternative to attention, utilizing extended convolutions and data-controlled gating mechanisms (Nguyen et al., 2024).
- DNABERT-2: As described in Section A.4.

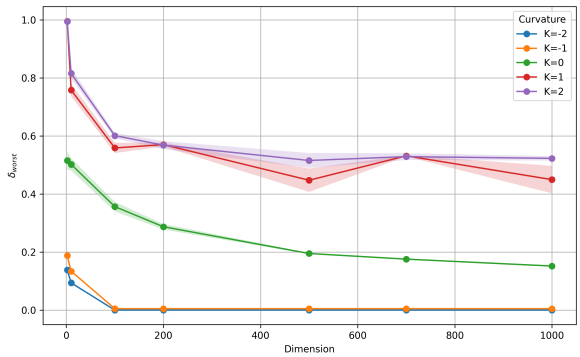

Figure 12: δ_{worst} estimates using simulated data points from the wrapped normal distribution on manifolds of varying curvatures (K) and dimensionalities.

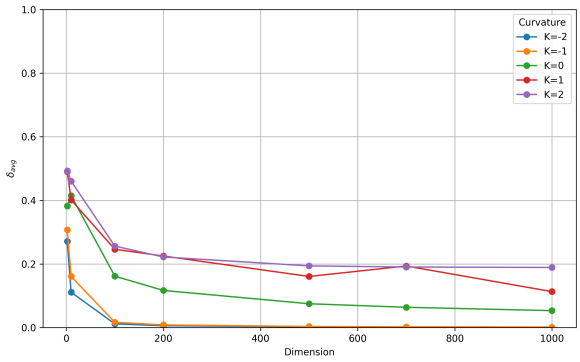

Figure 13: δ_{avg} estimates using simulated data points from the wrapped normal distribution on manifolds of varying curvatures (K) and dimensionalities.

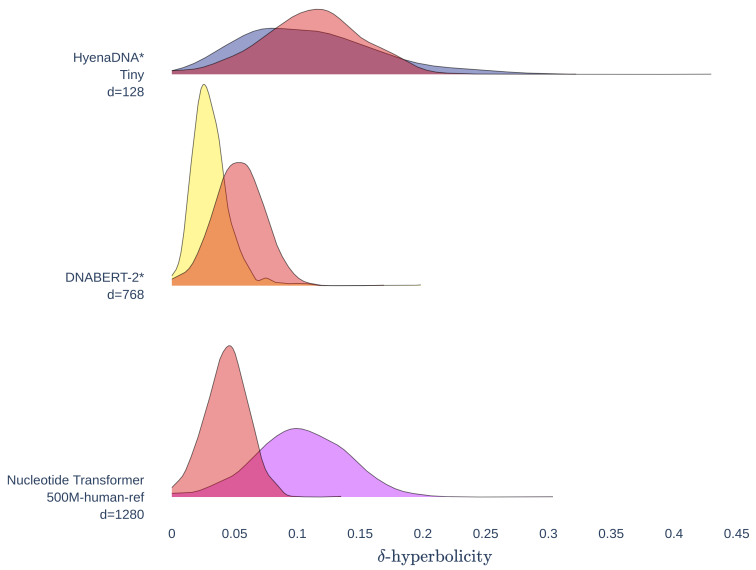

Figure 14: Distribution of scaled δ-hyperbolicity values using embeddings from various DNA LMs. The distribution of each model is overlaid with the the δ distribution of randomly sampled points on a Gaussian of equal dimensionality (red). * Denotes that the corresponding distribution constitutes smaller δ values (is more hyperbolic) than the Euclidean reference based on the Wilcoxon rank-sum test ($p < 0.01$)

- Nucleotide Transformer: A transformer-based model with 500 million parameters, trained on a comprehensive dataset comprising 3,202 human genomes and 850 genomes from diverse organisms (Lopez et al., 2023).

As a case study, we probe a subset of sequences that likely reflect strongly conserved evolutionary relationships. We therefore generate LM embeddings of a randomly sampled set of SINE sequences from TEB. The embeddings are derived by applying mean-pooling over the final layer embedding output of each model. To establish a comparative baseline, we juxtapose the underlying δ distribution of each LM with a distribution generated from randomly sampled points drawn from a Gaussian of equivalent dimensionality, following the procedure outlined in Section 5.2.

The results of our analysis are presented in Figure 14. Notably, we observe that the embeddings produced by HyenaDNA and DNABERT-2 exhibit significantly higher degrees of hyperbolicity compared to a null distribution of d-dimensional points ($p < 0.01$, Wilcoxon rank-sum test). In contrast, the representations generated by the Nucleotide Transformer demonstrate markedly lower hyperbolicity than the null distribution. This disparity may be attributed to the higher dimensionality of the Nucleotide Transformer embeddings, suggesting that the necessity for hyperbolic geometry may diminish as the latent space expands.

A.10 Hyperbolic Sequence Representations

In exploring the sequence representations of HCNNs, we started with the intuition built by Khrulkov et al. (2020), where hyperbolic image embeddings of MNIST near the center of the Poincaré disk represent the most ambiguous looking digits, while clear images lie near the boundary. Similarly, in Figure 15, we observe that in the processed pseudogene dataset in TEB, the sequence embeddings that lie close to the center of the Poincaré disk (the top of the hierarchy) correspond to low confidence embeddings for HCNNs (approximated by model loss on label predictions), while the embeddings near the disk boundaries show the highest classification confidence. This is consistent with the idea

Table 8: δ-Hyperbolicity values of the final embeddings for CNNs trained on each genomic dataset. Results are averaged over 10 sampling runs.

Benchmark	Task	Dataset	δ_{worst}	δ_{avg}
TEB	Retrotransposons	LTR Copia	$0.36_{\pm 0.0175}$	$0.145_{\pm 0.0019}$
		LINEs	$0.40_{\pm 0.0110}$	$0.164_{\pm 0.0004}$
		SINEs	$0.08_{\pm 0.0076}$	$0.170_{\pm 0.0016}$
	DNA transposons	CMC-EnSpm	$0.18_{\pm 0.0181}$	$0.163_{\pm 0.0009}$
		hAT-Ac	$0.37_{\pm 0.0220}$	$0.215_{\pm 0.0026}$
	Pseudogenes	processed	$0.36_{\pm 0.0204}$	$0.189_{\pm 0.0007}$
		unprocessed	$0.35_{\pm 0.0140}$	$0.157_{\pm 0.0003}$
GUE	Epigenetic Marks Prediction	H3	$0.10_{\pm 0.0072}$	$0.098_{\pm 0.0005}$
		H3K14ac	$0.09_{\pm 0.0090}$	$0.101_{\pm 0.0030}$
		H3K36me3	$0.26_{\pm 0.0541}$	$0.251_{\pm 0.0014}$
		H3K4me1	$0.21_{\pm 0.0185}$	$0.225_{\pm 0.0056}$
		H3K4me2	$0.13_{\pm 0.0112}$	$0.125_{\pm 0.0039}$
		H3K4me3	$0.14_{\pm 0.0168}$	$0.169_{\pm 0.0020}$
		H3K79me3	$0.15_{\pm 0.0255}$	$0.122_{\pm 0.0067}$
		H3K9ac	$0.21_{\pm 0.0160}$	$0.265_{\pm 0.0058}$
		H4ac	$0.18_{\pm 0.0156}$	$0.186_{\pm 0.0024}$
		H4	$0.10_{\pm 0.0058}$	$0.082_{\pm 0.0041}$
	Human Transcription Factor Prediction	0	$0.20_{\pm 0.0114}$	$0.160_{\pm 0.0026}$
		1	$0.20_{\pm 0.0245}$	$0.152_{\pm 0.0044}$
		2	$0.19_{\pm 0.0189}$	$0.148_{\pm 0.0021}$
		3	$0.19_{\pm 0.0189}$	$0.141_{\pm 0.0004}$
		4	$0.18_{\pm 0.0098}$	$0.140_{\pm 0.0009}$
	Splice Site Prediction	splice	$0.29_{\pm 0.0363}$	$0.256_{\pm 0.0012}$
	Mouse Transcription Factor Prediction	0	$0.21_{\pm 0.0147}$	$0.140_{\pm 0.0043}$
		1	$0.35_{\pm 0.0301}$	$0.249_{\pm 0.0032}$
		2	$0.21_{\pm 0.0226}$	$0.139_{\pm 0.0011}$
		3	$0.19_{\pm 0.0237}$	$0.131_{\pm 0.0009}$
		4	$0.19_{\pm 0.0112}$	$0.148_{\pm 0.0022}$
	Covid Variant Classification	covid	$0.50_{\pm 0.0388}$	$0.417_{\pm 0.0030}$
	Core Promoter Detection	all	$0.29_{\pm 0.0105}$	$0.229_{\pm 0.0034}$
		notata	$0.28_{\pm 0.0184}$	$0.212_{\pm 0.0010}$
		tata	$0.22_{\pm 0.0082}$	$0.138_{\pm 0.0013}$
	Promoter Detection	all	$0.29_{\pm 0.0146}$	$0.260_{\pm 0.0024}$
		notata	$0.31_{\pm 0.0210}$	$0.257_{\pm 0.0043}$
		tata	$0.16_{\pm 0.0127}$	$0.138_{\pm 0.0069}$
GB	Demo	coding vs intergenomic seqs	$0.21_{\pm 0.0180}$	$0.118_{\pm 0.0019}$
		human or worm	$0.19_{\pm 0.0189}$	$0.121_{\pm 0.0010}$
	Enhancers	drosophila enhancers stark	$0.30_{\pm 0.0174}$	$0.209_{\pm 0.0012}$
		human enhancers cohn	$0.19_{\pm 0.0137}$	$0.092_{\pm 0.0002}$
		human enhancers ensembl	$0.19_{\pm 0.0198}$	$0.109_{\pm 0.0001}$
	Regulatory	human ensembl regulatory	$0.23_{\pm 0.0282}$	$0.148_{\pm 0.0013}$
		human non-tata promoters	$0.19_{\pm 0.0053}$	$0.103_{\pm 0.0002}$
	Open Chromatin Regions	human ocr ensembl	$0.24_{\pm 0.0400}$	$0.189_{\pm 0.0011}$

that well defined sequences are at the bottom of the hierarchy where there is more space to separate out nuanced differences between sequences based on distinctive features.

Next, we conduct an experiment to dissect the sequence features informing the hyperbolic genome embedding. Given our dataset of processed pseudogenes, we examine the changes made to the HCNN representation by perturbing a fixed pseudogene sequence. For a fixed sequence, we follow these steps:

1. Compute the Genomic Evolutionary Rate Profiling (GERP) Cooper et al. (2005) score for each nucleotide along the sequence. GERP scores quantify evolutionary constraints at specific genomic positions, identifying which positions are functionally important based on selective pressure. GERP uses multiple sequence alignments across species to identify conserved regions.
2. Mutate a fraction of the nucleotides under the highest selective pressure (repeat for multiple perturbed sequences).
3. Use HCNN to generate an embedding for this perturbed instance of our original sequence.

Figure 16 visualizes this experiment using a processed pseudogene sequence and a background sequence. As the evolutionary signal under strong selection is eroded by the introduced mutations, it is likely that the features that make the pseudogene more "gene-like" are degraded. This degradation ultimately makes the sequence more ambiguous to the HCNN, and we observe that the perturbed representations move closer to the top of the hierarchy (near the center of the Poincaré disk), where the low confidence sequences lie. Removal of these evolutionary features actively hinders the model in identifying pseudogenes. However, perturbing conserved regions from the noisy background sequences does not appear to have this effect, as the model focuses on learning features common to the pseudogene class.

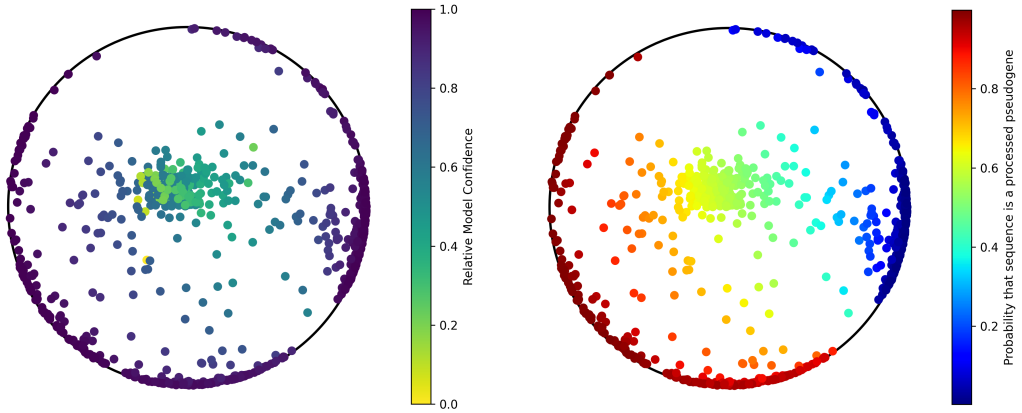

Figure 15: HCNN embeddings for the processed pseudogene dataset, colored by model confidence on the left, and by the probability that the sequence is a processed pseudogene (vs. a background sequence) on the right. Sequences embeddings are visualized on the Poincaré disk.

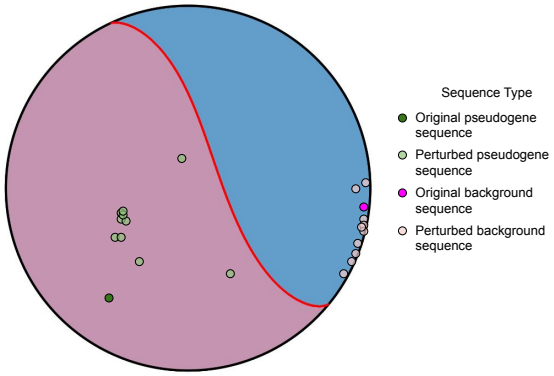

Figure 16: HCNN embeddings for a processed pseudogene sequence and background sequence. Each sequence has been perturbed multiple times, with different instances shown on the Poincaré disk.