HYPERBOLIC ASSOCIATIVE MEMORY NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Associative memory models encode a set of candidate patterns as "memories" and, upon receiving a partial or noisy query, retrieve the patterns most relevant to the query via similarity interactions/energy minimization, thereby recovering or recalling target patterns from incomplete inputs; they have achieved widespread success across many perception and representation learning tasks. However, when the retrieval process is constrained to Euclidean geometry, hierarchical structure in the data is difficult to capture accurately: in many tasks that require handling hierarchical data, Hopfield networks based on Euclidean representations tend to introduce bias and distortion into semantic relations. To this end, we extend modern Hopfield retrieval to hyperbolic space. Specifically, we map query and memory vectors from Euclidean space to hyperbolic space via exponential maps, and define an energy function with clear theoretical grounding based on the Minkowski inner product; the retrieval procedure adopts Riemannian manifold optimization, combining curvature-aware gradients with exponential maps to ensure that the optimization trajectory remains on the manifold and yields stable updates. Our central view can be stated as a hierarchy-sensitivity hypothesis: when the data exhibit clear and deeper hierarchical structure, hyperbolic geometry brings statistically significant improvements; when the hierarchy is weak or only shallow, performance shows no significant difference from Euclidean modern Hopfield networks. We validate this through depth-controlled comparisons and cross-level consistency metrics, and the empirical results are consistent with the hypothesis. Accordingly, the proposed hyperbolic associative memory can serve as a plug-and-play general memory module embedded into task architectures that require hierarchical understanding, for storing and retrieving raw inputs, intermediate representations, or learned prototypes, and explicitly exploiting hierarchical information. Moreover, our method is formulated in a model-agnostic manner and applies to any hyperbolic model with constant negative curvature. In this paper, we instantiate it with the Poincaré ball for experiments.

1 INTRODUCTION

Associative memory models, such as Hopfield networks, have played a crucial role in enabling neural systems to retrieve stored patterns from partial or noisy inputs. In this domain, classical Hopfield network models Hopfield (1982); Amari (1972) store memories as fixed-point attractor states in an energy landscape, leveraging Hebbian learning to recall full patterns from partial input cues through a recurrent architecture. More recently, Modern Hopfield Networks (MHNs) Vaswani et al. (2017); Widrich et al. (2020) have introduced continuous relaxations of the original formulation, theoretically achieving exponential storage capacity with respect to the number of neurons Krotov & Hopfield (2016); Demircigil et al. (2017); Ramsauer et al. (2020) and reigniting interest in associative memory mechanisms. MHNs have been successfully applied to tasks such as immune repertoire classification and graph anomaly detection Hoover et al. (2023).

While representing data in Euclidean space $\mathbb{R}^n$ has long been the standard choice due to its computational convenience—providing closed-form expressions for distances, inner products, and straightforward input into neural networks, recent studies have revealed fundamental limitations of this approach for complex data types Ganea et al. (2018a). Many real-world datasets, particularly those involving graphs, taxonomies, or hierarchical relationships, exhibit an inherently non-Euclidean latent structure Bronstein et al. (2017). In such cases, Euclidean embeddings often struggle to faithfully

preserve semantic proximity and hierarchical organization Gromov (1987). For example, arbitrary tree structures cannot be embedded with arbitrarily low distortion even in high-dimensional Euclidean spaces Linial et al. (1995), whereas hyperbolic spaces, owing to their exponential growth of volume, can naturally accommodate such structures even in low dimensions Krioukov et al. (2010); Nickel & Kiela (2017). Thus, in tasks of this kind (e.g. hierarchical classification, hierarchical clustering, knowledge graph completion, and graph/image/text classification or retrieval with hierarchical labels), applying associative memory mechanisms purely within Euclidean geometry may distort the underlying structural information during memory retrieval. These observations motivate us to embed the associative memory process into hyperbolic space—which is naturally suited to representing hierarchical and structured information.

To address these limitations, we introduce Hyperbolic Associative Memory Networks (HAMNs), the first framework that embeds modern associative memory into hyperbolic space. Specifically, we first apply exponential maps to transform query and memory vectors from Euclidean space to hyperbolic space (a constant–negative–curvature manifold), thereby leveraging the natural capacity of hyperbolic geometry to model hierarchical structures. On top of these mapped representations, we define a principled energy function using the Minkowski inner product to capture similarity relations in hyperbolic geometry. During memory retrieval, we incorporate curvature-aware Riemannian optimization Bonnabel (2013) with exponential-map updates to ensure that each update step follows the tangent direction of the hyperbolic manifold and remains strictly within hyperbolic space. In our experiments, we instantiate the method with the Poincaré ball Nickel & Kiela (2017) due to implementation maturity, while the derivations apply equally to other hyperbolic models (e.g. Lorentz, Klein).

With this design, we propose a hierarchy-sensitivity hypothesis that does not presuppose pronounced hierarchical structure in all tasks or datasets; when hierarchical/tree structure does exist and is sufficiently deep, HAMNs demonstrate a stronger ability to understand, preserve, and retrieve hierarchical relations, whereas when the hierarchy is weak or essentially absent, their performance is largely on par with Euclidean MHNs. To validate this hypothesis, we conduct a systematic evaluation by controlling hierarchy depth and reporting metrics such as cross-level consistency, and the empirical results are consistent with the hypothesis.

Our main contributions are summarized as follows:

- We design Hyperbolic Associative Memory Networks (HAMNs), a plug-and-play, model-agnostic associative memory module operating in hyperbolic space that can be dropped into architectures requiring hierarchical understanding to store and retrieve raw inputs, intermediate representations, or learned prototypes, explicitly leveraging hierarchical structure.
- We design a principled energy function and optimization mechanism based on hyperbolic geometry, ensuring a stable and efficient memory update process.
- With hierarchy depth controlled and cross-level consistency measured, our method achieves clear benefits on hierarchical data and competitive flat/shallow results, outperforming Euclidean Hopfield networks at representing complex structures.

2 PRELIMINARIES

2.1 MODERN HOPFIELD NETWORKS

Modern Hopfield Networks (MHNs) Krotov & Hopfield (2016); Demircigil et al. (2017); Ramsauer et al. (2020) extend classical associative memory models by introducing continuous state representations and modifying the energy function landscape. This modification significantly enhances the storage capacity and enables the network to retrieve stored patterns through continuous optimization dynamics.

Given a set of N memory patterns $\{\xi_n \in \mathbb{R}^K\}_{n=1}^N$, organized as a memory matrix $\Xi \in \mathbb{R}^{N\times K}$, and a query state vector $s \in \mathbb{R}^K$, the energy function of MHNs is formulated as:

$$E(s, \Xi; \beta) = F_\beta\left(f_{\text{sim}}(\{\xi_n\}, s)\right) + \frac{1}{2} s^\top s \tag{1}$$

where the similarity is defined as $f_{\text{sim}}(\{\xi_n\}, s) = \{\langle \xi_n, s \rangle\}_{n=1}^N$ (dot product between s and each memory), and $F_\beta(\cdot)$ is the log-sum-exponential (LSE) function:

$$F_\beta(z) = -\frac{1}{\beta} \log \left(\sum_{n=1}^{N} \exp(\beta z_n) \right) \tag{2}$$

with $\beta > 0$ controlling the sharpness. For consistency with the rest of the paper, we will use θ as the temperature (i.e., $\theta \equiv \beta$).

The associative retrieval process minimizes the energy iteratively as:

$$s^{(t+1)} = \Xi \,\text{softmax}\left(\beta \Xi^\top s^{(t)}\right) = \sum_{n=1}^{N} \xi_n \frac{\exp\left(\beta \xi_n^\top s^{(t)}\right)}{\sum_{n'=1}^{N} \exp\left(\beta \xi_{n'}^\top s^{(t)}\right)} \tag{3}$$

Under mild conditions, the update rule monotonically decreases the system energy and converges to a (meta-)stable fixed point Ramsauer et al. (2020); Widrich et al. (2020). This framework thus enables efficient pattern retrieval even from noisy or partial cues. Eq. (3) is equivalent to the readout of single-head attention, with keys=values $= X$ and query $s^{(t)}$, hence an MHN can be viewed as an energy-based realization of attention.

2.2 Hyperbolic manifolds: concepts and intuition

A **hyperbolic manifold** is a Riemannian manifold $(\mathcal{M}, g)$ Cannon et al. (1997) of *constant negative curvature* $-c < 0$. Geometrically it exhibits:

- Triangle angle sum $< \pi$; geodesics diverge; ball volume grows **exponentially** with radius (matching the exponential branching of trees/hierarchies).
- Any two points are typically joined by a unique geodesic; distances near the boundary are "magnified".
- Multiple *isometric* coordinate models (Poincaré ball/upper-half plane, Klein, Lorentz hyperboloid) that are mutually isometric and differ only by parametrization.

Our theory and algorithm rely only on *model-agnostic* primitives; a concrete instantiation (e.g. Poincaré ball) is deferred to implementation details.

2.2.1 Primitive 1: exponential/logarithmic maps

Definition. For any $p \in \mathcal{M}$, the exponential map

$$\exp_p^c : T_p\mathcal{M} \to \mathcal{M}, \qquad \exp_p^c(v) = \gamma_v(1)$$

sends a tangent vector v to the point at unit time on the geodesic γ_v starting at p with initial velocity v. The logarithmic map $\log_p^c : \mathcal{M} \to T_p\mathcal{M}$ is the local inverse of $\exp_p^c$ around p.

Properties.

1. $\exp_p^c(0) = p$ and $\mathrm{d}(\exp_p^c)_0 = \mathrm{id}$;
2. for small steps, $\exp_p^c(v)$ is locally $p + v$ in coordinates;
3. Exp/Log provide a two-way bridge between the Euclidean tangent space and the manifold, enabling *on-manifold* encoding/optimization.

Algorithmic use. Map Euclidean queries/memories into $T_0\mathcal{M}$ and then onto the manifold via $\exp_0^c$; at iteration t, compute a descent direction in $T_{\xi^t}\mathcal{M}$ and return to the manifold with $\exp_{\xi^t}^c$.

2.2.2 Primitive 2: geodesic distance

Definition. $d_{\mathcal{M}}(x, y)$ is the length of the shortest geodesic between x and y induced by g.

Hierarchy intuition. Radial distance grows roughly linearly with radius, but near the boundary any fixed Euclidean displacement is exponentially magnified, naturally separating differences in hierarchical depth (see the toy example 2.2.5).

2.2.3 PRIMITIVE 3: "MINKOWSKI-LIKE" INNER PRODUCT

We use

$$\boxed{\langle x, y\rangle_M \;:=\; -\cosh\big(d_{\mathcal{M}}(x,y)\big)}$$

as the similarity in hyperbolic space. Key properties:

- **Monotonicity:** strictly decreasing in $d_{\mathcal{M}}(x,y)$; equals -1 at $x = y$ and tends to $-\infty$ as distance increases.
- **Equivalence:** in the Lorentz model this similarity is a monotone function of the Minkowski bilinear form; it coincides numerically with it when curvature is -1. This choice is model-agnostic and invariant under hyperbolic isometries.

2.2.4 ISOMETRY INVARIANCE

If $\phi : (\mathcal{M}, g) \to (\mathcal{M}', g')$ is an isometry, then

$$d_{\mathcal{M}}(x,y) = d_{\mathcal{M}'}(\phi(x),\phi(y)) \quad \Rightarrow \quad \langle x, y\rangle_M = \langle \phi(x), \phi(y)\rangle_{M'}.$$

Hence, any energy and update constructed from $d_{\mathcal{M}}$ and $-\cosh d_{\mathcal{M}}$ are *model-equivalent* across hyperbolic realizations (Poincaré ball, Lorentz, Klein, upper-half plane, etc.).

2.2.5 WHY HYPERBOLIC FOR HIERARCHIES? A TOY EXAMPLE

For simplicity in this toy example we take curvature $-c = -1$ (i.e., $c = 1$) on the Poincaré ball.

Euclidean space "flattens" hierarchies. Embed a tree of depth L into the plane: all nodes at level ℓ lie on the same-radius circle. As L grows, the outermost nodes crowd the same ring and leaf–leaf distances are governed almost only by the angular gap and become very similar, so leaves from different major branches appear "about equally far" and hierarchical information is weakened.

Hyperbolic space "pulls apart" hierarchies. Keep angles uniform, but encode depth by hyperbolic radius:

$$\rho_\ell = \tanh(\alpha\ell/2), \quad \alpha > 0.$$

Since $d(0,x) = 2\,\mathrm{artanh}\,\|x\|$, any level-$\ell$ node satisfies:$d_{\mathbb{D}}(0, x_\ell) = \alpha\,\ell$

i.e., each additional level increases the *radial hyperbolic distance* by (approximately) a fixed amount, so different levels separate naturally. Moreover, because the metric is "magnified" near the boundary, two points on the *same level* but from *different major branches* acquire a much larger hyperbolic distance even for a tiny angular gap, whereas points within the same subtree are closer.

Rule of thumb (consistency with hierarchy). If two leaves have lowest common ancestor depth a, then the dominant term of their distance is

$$d_{\mathbb{D}}(x_i, x_j) \;\approx\; 2\alpha\,(L - a)\;(+\text{ lower-order terms}),$$

which increases strictly with tree distance and is monotone in the LCA depth ("closer relatives" are more similar). Hence hyperbolic space simultaneously preserves two key signals—*depth* (radial) and *branching relation* (angular)—and avoids the hierarchical "flattening" of Euclidean embeddings.

3 METHODOLOGY

Our proposed Hyperbolic Associative Memory Networks (HAMNs) use hyperbolic geometry to store and retrieve patterns. This section introduces the core components of HAMNs.

3.1 MEMORY ENCODING IN HYPERBOLIC SPACE

We first map all memories and the query onto a common hyperbolic manifold. Let $x_i^R \in \mathbb{R}^d$ $(i = 1, \ldots, N)$ denote the N stored patterns in Euclidean space (these can be regarded as the keys in memory), and let $\xi^R \in \mathbb{R}^d$ be the query pattern (the cue or initial state). Let $(\mathcal{M}, g)$

be a complete, simply connected Riemannian manifold with constant negative curvature $-c < 0$. Choose a reference point $p \in \mathcal{M}$ and fix an orthonormal frame on its tangent space $T_p\mathcal{M}$, thereby identifying $T_p\mathcal{M} \cong \mathbb{R}^d$ via an isometric isomorphism $\iota_p : \mathbb{R}^d \to T_p\mathcal{M}$. We encode using the exponential map at p:

$$v_i = \iota_p(x_i^R), \qquad v_\xi = \iota_p(\xi^R), \qquad x_i = \exp_p^c\big(\Pi(v_i)\big), \quad \xi = \exp_p^c\big(\Pi(v_\xi)\big), \tag{4}$$

To avoid, in some models, mapped points becoming too close to the boundary (which may lead to numerical instability and gradient explosion), we may perform norm clipping in the tangent space *before* the exponential map. Given a clipping threshold $\texttt{clip}_{\tan} > 0$, $\Pi(\cdot)$ denotes tangent-space norm clipping:

$$\Pi(v) \;=\; v \cdot \min\left(1,\, \frac{\texttt{clip}_{\tan}}{\|v\| + \varepsilon}\right), \qquad \varepsilon > 0. \tag{5}$$

Here ε is a small constant for numerical stability (e.g. 10^{-5}). After this encoding step, all memory points x_i and the query point ξ lie on the manifold $\mathcal{M}$.

3.2 Energy Function Design

On a hyperbolic manifold, we use an energy function $E(\xi)$ to measure how well the current retrieval state ξ matches the stored patterns $\{x_i\}_{i=1}^N$: the energy should be low when ξ is close to some memory x_i, and high otherwise. To this end, we replace the Euclidean inner product by a *hyperbolic similarity*:

$$\langle x, y\rangle_M \;:=\; -\cosh\big(d_{\mathcal{M}}(x, y)\big),$$

where $d_{\mathcal{M}}$ is the geodesic distance induced by the metric g. This similarity is identical across hyperbolic models; in particular, in the Lorentz (hyperboloid) model $\langle x, y\rangle_M$ coincides with the classical Minkowski inner product, while in other models it can be computed directly from $d_{\mathcal{M}}$ without explicitly mapping between models.

Accordingly, for any $\xi \in \mathcal{M}$ we define the energy as

$$E(\xi) \;=\; -\frac{1}{\theta}\log\left(\sum_{i=1}^{N}\exp\big(\theta\,\langle x_i, \xi\rangle_M\big)\right) \;+\; \frac{1}{2}\,d_{\mathcal{M}}(\xi, p)^2, \tag{6}$$

where $\theta > 0$ is a temperature parameter and $p \in \mathcal{M}$ is a fixed reference point (e.g., the origin in Poincaré coordinates). We use the *intrinsic* squared geodesic regularizer $\frac{1}{2}d_{\mathcal{M}}(\xi, p)^2$, which is geodesically convex in hyperbolic space and penalizes deviations from p.

The first term in equation 6 is a smooth approximation to the "maximum similarity": when θ is large, $-\frac{1}{\theta}\log\sum_i \exp(\theta\langle x_i, \xi\rangle_M) \;\approx\; -\max_i\langle x_i, \xi\rangle_M$, so it is minimized when ξ is close to one of the memories x_i. The second term penalizes large geodesic deviations from p, suppressing excursions toward the boundary and stabilizing the optimization trajectory.

Together, these two terms yield energy minima around stored memories. When $\xi = x_k$, we have $d_{\mathcal{M}}(x_k, \xi) = 0$ and $\langle x_k, \xi\rangle_M = -1$, leading to a low energy; conversely, when ξ is far from all memories, the energy becomes large. Further discussion of energy bounds is provided in Appendix A.1. For a detailed discussion of storage capacity, see Appendix B.

3.3 Memory Retrieval and Optimization

We optimize the retrieval energy using the *Concave–Convex Procedure (CCCP)* Yuille & Rangarajan (2001). A detailed derivation for our setting is provided in Appendix A.2; here we summarize the resulting update rules.

CCCP decomposition. Decompose $E(\xi)$ in Eq. equation 6 into a geodesically convex term and a concave term on a Hadamard manifold:

$$E(\xi) \;=\; E_{\text{cvx}}(\xi) + E_{\text{cave}}(\xi), \qquad E_{\text{cvx}}(\xi) = \tfrac{1}{2}\,d_{\mathcal{M}}(\xi, p)^2, \quad E_{\text{cave}}(\xi) = -\frac{1}{\theta}\log\Big(\sum_{i=1}^{N} e^{\,\theta\,\langle x_i, \xi\rangle_M}\Big), \tag{7}$$

where $p \in \mathcal{M}$ is a fixed reference point and $\langle x, \xi\rangle_M := -\cosh(d_{\mathcal{M}}(x, \xi))$ denotes the hyperbolic similarity. The squared distance is geodesically convex on Hadamard manifolds.

Softmax weights. At iteration t, define

$$p_i^{(t)} \;=\; \frac{\exp\big(\theta\,\langle x_i, \xi^{(t)}\rangle_M\big)}{\sum_{j=1}^{N}\exp\big(\theta\,\langle x_j, \xi^{(t)}\rangle_M\big)}. \tag{8}$$

Riemannian linearization and surrogate. Let $a^{(t)} := \operatorname{grad} E_{\mathrm{cave}}(\xi^{(t)})$ be the *Riemannian* gradient at $\xi^{(t)}$. The concave part admits the first-order (Riemannian) upper bound

$$E_{\mathrm{cave}}(\xi) \;\le\; E_{\mathrm{cave}}(\xi^{(t)}) \;+\; \big\langle a^{(t)},\, \log_{\xi^{(t)}}(\xi)\big\rangle_{\xi^{(t)}},$$

so the CCCP surrogate reads

$$Q\Big(\xi \mid \xi^{(t)}\Big) \;=\; \tfrac{1}{2}\, d_{\mathcal{M}}(\xi, p)^2 \;+\; \big\langle a^{(t)},\, \log_{\xi^{(t)}}(\xi)\big\rangle_{\xi^{(t)}} \quad \text{(constants dropped)}. \tag{9}$$

Closed-form CCCP step. The minimizer of equation 9 on a Hadamard manifold is obtained in closed form:

$$\xi^{(t+1)} \;=\; \exp_p\Big(\,-\,\mathrm{PT}_{\xi^{(t)}\to p}\big(a^{(t)}\big)\Big), \tag{10}$$

where $\mathrm{PT}_{\xi^{(t)}\to p}$ denotes parallel transport along the unique geodesic from $\xi^{(t)}$ to p. Equivalently, introducing $v^{(t)} := -\,\mathrm{PT}_{\xi^{(t)}\to p}\big(a^{(t)}\big)$ and a damping step size $\eta \in (0, 1]$, we use the stable update

$$\xi^{(t+1)} \;=\; \exp_p\big(\,\eta\, v^{(t)}\,\big), \tag{11}$$

for which $\eta = 1$ recovers the exact minimizer in equation 10.

Intrinsic gradient. Using equation 8, the Riemannian gradient of the concave term can be written as

$$a^{(t)} \;=\; \operatorname{grad} E_{\mathrm{cave}}(\xi^{(t)}) \;=\; -\sum_{i=1}^{N} p_i^{(t)}\, \operatorname{grad}_{\xi}\,\langle x_i, \xi\rangle_M\Big|_{\xi=\xi^{(t)}}. \tag{12}$$

Convergence note. Since E_{cvx} is geodesically convex and E_{cave} is concave, the CCCP iterations monotonically decrease $E(\xi)$ on Hadamard manifolds; the sequence $\{\xi^{(t)}\}$ converges to a (meta-)stable fixed point corresponding to a stored memory.

3.4 Hyperbolic Hopfield Modules for Deep Learning

Inspired by the modular design of modern Hopfield networks Ramsauer et al. (2020), we adopt a similar architecture for modularization and replace its original Euclidean update mechanism with the hyperbolic retrieval strategy proposed in this paper. Based on this formulation, we construct three core modules—*Hyperbolic Hopfield* (**HypHopfield**), *Hyperbolic Hopfield Pooling* (**HypPooling**), and *Hyperbolic Hopfield Layer* (**HypLayer**)—targeting association, aggregation, and retrieval, respectively. All three are implemented on the Poincaré ball model and can be seamlessly integrated into deep neural networks, thereby enhancing hierarchical modeling and memory capabilities. Detailed structure and implementation are provided in Appendix D.

4 Experiments

Instantiation. All experiments instantiate HAMNs on the Poincaré ball model (constant negative curvature $-c$); the model-agnostic derivations hold for any hyperbolic realization, and concrete formulas for instantiations on common hyperbolic models are provided in Appendix C.

Overview We systematically evaluate HAMNs around the "hierarchy-sensitivity hypothesis" and delineate their effectiveness and scope via four groups of experiments:

(i) **CIFAR-100 hierarchical classification**: On our 2/3/4-layer label trees, **HAMNs** deliver the strongest cross-level consistency and competitive accuracy across levels, clearly outperforming **Euclidean MHNs**; the consistency gap widens as the hierarchy deepens. **HypAttn** is strongest for shallow/mid-level retrieval, while **HypNN** excels at fine-grained recognition.

Table 1: Hierarchical classification on CIFAR-100 results.

Model	top_acc	super_acc	coarse_acc	fine_acc	coph_corr
CIFAR-100-2-layer					
Backbone only	—	—	64.20 ± 0.91	51.00 ± 1.28	0.6652 ± 0.0163
HypAttn	—	—	$\mathbf{70.67 \pm 0.56}$	$\mathbf{58.19 \pm 0.38}$	0.6740 ± 0.0142
HypNN	—	—	69.02 ± 0.47	56.82 ± 0.41	0.5938 ± 0.0202
MHNs	—	—	65.34 ± 0.86	49.86 ± 0.63	0.6295 ± 0.0143
HAMNs (ours)	—	—	70.12 ± 0.57	56.00 ± 0.64	$\mathbf{0.6778 \pm 0.0193}$
CIFAR-100-3-layer					
Backbone only	—	72.75 ± 1.89	62.58 ± 1.35	50.79 ± 0.82	0.7023 ± 0.0164
HypAttn	—	79.33 ± 0.66	68.68 ± 0.78	54.01 ± 0.97	0.6902 ± 0.0240
HypNN	—	79.40 ± 0.66	$\mathbf{68.84 \pm 0.95}$	54.09 ± 1.05	0.7123 ± 0.0211
MHNs	—	79.17 ± 0.59	68.08 ± 0.84	52.89 ± 0.97	$\mathbf{0.7152 \pm 0.0256}$
HAMNs (ours)	—	$\mathbf{79.70 \pm 0.29}$	68.81 ± 0.59	$\mathbf{54.27 \pm 0.47}$	0.7017 ± 0.0658
CIFAR-100-4-layer					
Backbone only	87.51 ± 0.73	72.68 ± 1.85	60.02 ± 1.02	47.23 ± 0.77	0.7180 ± 0.0230
HypAttn	90.13 ± 0.48	78.23 ± 0.48	67.74 ± 0.93	54.50 ± 0.78	0.6795 ± 0.0143
HypNN	90.30 ± 0.35	78.72 ± 0.59	68.29 ± 0.80	$\mathbf{55.93 \pm 0.88}$	0.6046 ± 0.0149
MHNs	89.39 ± 0.29	76.97 ± 0.44	65.56 ± 0.42	49.37 ± 0.57	0.5902 ± 0.0218
HAMNs (ours)	$\mathbf{90.98 \pm 0.39}$	$\mathbf{79.48 \pm 0.57}$	$\mathbf{68.51 \pm 0.84}$	53.49 ± 1.05	$\mathbf{0.7184 \pm 0.0254}$

(ii) **Weak/shallow hierarchical tasks**: On classical MIL multi-instance learning and MoleculeNet molecular property prediction (where hierarchy is weak or only shallow), HAMNs perform on par with Euclidean MHNs overall, with slight advantages on a few datasets;

(iii) **Computation/performance comparison**: Theoretically fewer FLOPs and parameters, but due to hyperbolic operations and memory-access overhead, the current GPU implementation exhibits longer runtime and higher peak memory;

(iv) **Ablation studies**: Performance is best when the curvature c lies in a moderate range (approximately 0.7–2.0); using too many stored patterns degrades top-level accuracy, though the method is overall robust to this hyperparameter.

4.1 Hierarchical classification on CIFAR-100

To demonstrate that HAMNs can understand and exploit multi-level structure, we conduct hierarchical classification experiments on CIFAR-100. CIFAR-100 (60,000 color images of size 32×32) groups 100 *fine* classes into 20 *coarse* classes, yielding a balanced two-level hierarchy. Without modifying the original samples, we further cluster the 20 coarse classes into 7 "super" classes (e.g., large terrestrial vertebrates, plants, vehicles), and then group these 7 super classes into 3 "top" classes (animals, plants & natural scenes, man-made objects), thereby forming three- and four-level hierarchies.

On the model side, we adopt a ResNet-18 backbone with the final fully connected layer removed, and insert one of four memory/retrieval modules: (i) **HAMNs**[1], (ii) a hyperbolic attention baseline (**HypAttn**; (Gülçehre et al., 2019)), (iii) a lightweight hyperbolic neural block (**HypNN**; (Ganea et al., 2018a)), and (iv) Euclidean-space modern Hopfield networks (**MHNs**; (Ramsauer et al., 2020)). The retrieved representations are then fed into level-specific classification heads. We also report a **Backbone only** variant as a reference (See Table 1).

Coarse–Fine Coherence Correlation (`coph_corr`) measures the consistency between the model's "coarse" predictions and the "coarse" predictions obtained by aggregating its "fine" outputs.

From Table 1 we observe:

[1]Using our **HypLayer**; see Appendix D for details.

- **Euclidean MHNs:** Competitive but not leading overall. They achieve the best `coph_corr` on the 3-layer hierarchy, yet fall behind markedly on the deepest (4-layer) hierarchy in both high-level accuracies (top/super/coarse) and consistency; fine-grained accuracy is also below the hyperbolic baselines.
- **HypAttn vs. HypNN:** On shallow hierarchies (2-layer), HypAttn attains the highest `coarse_acc`/`fine_acc`. As depth increases, HypNN becomes strongest at fine-grained recognition (best `fine_acc` on 4-layer), while both methods lag behind HAMNs on top- and mid-level accuracies as well as cross-level coherence.
- **HAMNs (ours):** Improvements grow with depth. On 3-layer, HAMNs deliver the best `super_acc` and the best `fine_acc` with near-best `coarse_acc`. On 4-layer, they achieve state-of-the-art `top_acc`/`super_acc`/`coarse_acc` and the highest `coph_corr`; `fine_acc` remains competitive though below HypNN.

Takeaway. Hyperbolic, energy-based retrieval aligns predictions across hierarchy levels as depth grows. Euclidean MHNs can peak at a single level (e.g., 3-layer consistency) but do not scale to deeper hierarchies; HypAttn suits shallow aggregation, HypNN excels at fine granularity, and HAMNs strike the best accuracy–consistency balance overall.

4.2 Weak/shallow hierarchy tasks: MIL and molecular property prediction

Multi–Instance Learning (MIL). We evaluate on three classical MIL datasets—**Tiger**, **Elephant**, and **Fox**—to probe the bag–instance regime without instance-level labels (Dietterich et al., 1997), using the standard splits introduced by (Ilse et al., 2018; Küçükaşcı & Baydoğan, 2018; Carbonneau et al., 2018). We plug our **HypPooling** into the MIL pipeline: embedded instances serve as stored memories (Y), while a fixed set of learnable query vectors acts as state (query) patterns (R) on the same Poincaré ball; retrieval is performed via hyperbolic attention and on-manifold updates. See Appendix D for the layer design and Appendix E.1 for training protocol and hyperparameters. We compare against representative MIL baselines (e.g., attention-MIL (Ilse et al., 2018), mi-Net variants (Carbonneau et al., 2018), and Euclidean MHNs). Results show *competitive* overall performance and new SOTA on **Fox**; elsewhere the margins over Euclidean MHNs are modest (Table 3).

Molecular property prediction. Experiments on four MoleculeNet datasets—**HIV**, **BACE** (Subramanian et al., 2016), **BBBP** (Martins et al., 2012), and **SIDER** (Kuhn et al., 2016)—probe the weak/shallow–hierarchy regime. The proposed **HypLayer** is inserted into standard pipelines: training samples serve as stored memories (Y), inputs as queries (R), followed by hyperbolic embedding and retrieval (exact layer design, training protocol, and hyperparameters are detailed in Appendix D). Comparisons cover representative baselines (classical ML, GNNs, and Euclidean MHNs). This approach yields *competitive* overall performance and establishes new SOTA on **BBBP** and **SIDER** (full tables in Appendix E.2); nevertheless, margins over Euclidean MHNs remain small, consistent with the weak–hierarchy hypothesis.

4.3 Computational cost and performance

We compared our method against modern Hopfield networks in Euclidean space in terms of computational cost and performance. Using an input size of $128 \times 3 \times 224 \times 224$, in table 2.

Observations. From the table above, we observe: (i) **FLOPs And Parameter Count**—HAMNs require much fewer computations (27.2G FLOPs) and have far fewer parameters (3.3M) than Euclidean MHNs (108.3G FLOPs, 8.5M), making them theoretically more lightweight; (ii) **Runtime Overhead**—despite nearly $4\times$ fewer FLOPs, HAMNs are significantly slower in both for-

Table 2: Computational cost and performance comparison.

Method	FLOPs	Params	Forward + Backward	Forward Only	Peak GPU Mem
HAMNs	27.177 G	3.3 M	147.6 ms	43.0 ms	6578.8 MB
MHN	108.252 G	8.5 M	83.3 ms	23.4 ms	4305.3 MB

ward+backward (147.6ms vs. 83.3ms) and forward-only (43.0ms vs. 23.4ms) passes, reflecting the extra overhead of hyperbolic operations (e.g. Möbius addition, exponential/logarithmic maps, Riemannian gradient transforms) and associated memory-access costs; and (iii) **Memory Usage**—HAMNs consume more peak GPU memory (6.4GB) than MHNs (4.3GB), indicating that maintaining hyperbolic representations and intermediate states has a higher memory footprint.

4.4 Ablations: curvature and number of stored patterns

Summary. A *moderate curvature* provides the best trade-off across 2/3/4-level hierarchies; extremes are harmful (too small under-expresses hierarchy, too large degrades accuracy). Varying the *number of stored patterns* causes only small overall fluctuations: oversized memories reduce top-level accuracy, moderate increases help mid-level, and fine-level peaks at higher counts. In practice, we recommend *moderate curvature* and a *modest memory size*. See AppendixE.3 for details.

5 Related Work

Hopfield Networks. Hopfield networks were initially proposed by Hopfield (1982) as a type of recurrent neural network designed to store discrete binary patterns as stable attractors and to retrieve them via energy minimization dynamics, enabling associative memory functionality. To better handle continuous data, Tank & Hopfield (1986) introduced the continuous Hopfield network, which extends the state space from binary to real-valued domains. In recent years, modern Hopfield networks have advanced rapidly. By introducing differentiable continuous energy functions, they significantly improve memory capacity and support one-step convergence. A representative work is the modern Hopfield layer proposed by Ramsauer et al. (2021), which is highly compatible with deep learning models and can be viewed as a generalization of the attention mechanism. Building on prior work, we extend Hopfield networks to hyperbolic space to better model hierarchies.

Hyperbolic Geometry. Nickel & Kiela (2017) first proposed using hyperbolic space to learn hierarchical representations of symbolic data, such as text and graphs, by embedding them into the Poincaré ball model. Since then, the application of hyperbolic geometry has been explored in various domains. Ganea et al. (2018b) introduced hyperbolic neural network layers, which have enabled the development of hybrid architectures such as hyperbolic convolutional neural networks (Shimizu et al., 2021), hyperbolic graph convolutional networks (Chami et al., 2019), hyperbolic variational autoencoders (Ovinnikov et al., 2021), and hyperbolic attention networks (Gulcehre et al., 2019). These architectures have been successfully applied to tasks such as deep metric learning, object detection , and natural language processing. Beyond practical applications, theoretical investigations into hyperbolic spaces and their models have also deepened, demonstrating properties such as lower representation distortion (De Sa et al., 2018), better generalization ability (Bachmann et al., 2021), and stronger representation power in low-dimensional spaces (Sala et al., 2018). Unlike prior implicit uses of hyperbolic geometry, energy-based Hopfield retrieval is carried out directly in hyperbolic space, broadening applicability to hierarchical representation learning.

6 Discussion and Conclusion

We propose a plug-and-play, *model-agnostic* memory framework that generalizes modern Hopfield networks from Euclidean to hyperbolic geometry, formulating retrieval as energy minimization based on geodesic distance and its induced "Minkowski-like" similarity. As a general-purpose memory module, HAMNs can be deployed in any downstream task that requires storing and retrieving hierarchical patterns, providing a geometry-aware memory pathway for hierarchical modeling. Our experiments support the *hierarchy-sensitivity* hypothesis: as hierarchical depth increases, HAMNs deliver statistically significant gains; in flat or shallow settings, they perform on par with Euclidean MHNs. Importantly, hierarchy restructuring on CIFAR-like data serves as a *component-level* validation: by keeping the data fixed and altering only the label hierarchy, we can more cleanly test whether the memory module truly exploits hierarchical geometry. Compute/perf analysis shows a trade-off: despite lower theoretical FLOPs/parameters, current GPU hyperbolic ops and memory access add overhead, yielding longer runtimes and higher peak memory. Ablation studies show the model is overall robust to curvature and memory size.

REPRODUCIBILITY STATEMENT

We provide a runnable implementation of HAMNs instantiated on the Poincaré ball and the CIFAR-100 hierarchical experiments, submitted as supplementary materials. Code-level implementation details for **HypHopfield**, **HypPooling**, and **HypLayer** are given in Appx. D. For other common hyperbolic models (Lorentz, Klein, upper half-plane, hemisphere), model-agnostic replacement formulas are provided in Appx. C. The supplementary code package includes the training scripts and module-instantiation code required to reproduce the experiments.

REFERENCES

Shun-Ichi Amari. Learning patterns and pattern sequences by self-organizing nets of threshold elements. *IEEE Transactions on Computers*, 100(11):1197–1206, 1972.

Stuart Andrews, Ioannis Tsochantaridis, and Thomas Hofmann. Support vector machines for multiple-instance learning. *Advances in neural information processing systems*, 15, 2002.

Gregor Bachmann, Maximilian Nickel, and Mathias Niepert. Constant curvature manifolds for open set learning. *Advances in Neural Information Processing Systems*, 34, 2021.

Silvere Bonnabel. Stochastic gradient descent on riemannian manifolds. *IEEE Transactions on Automatic Control*, 58(9):2217–2229, 2013.

Michael M Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, 2017.

James W Cannon, William J Floyd, Richard Kenyon, Walter R Parry, et al. Hyperbolic geometry. *Flavors of geometry*, 31(59-115):2, 1997.

Marc-André Carbonneau, Veronika Cheplygina, Eric Granger, and Ghyslain Gagnon. Multiple instance learning: A survey of problem characteristics and applications. *Pattern recognition*, 77: 329–353, 2018.

Ines Chami, Aditya Wolf, Frederic Sala, Christopher Re, and Jure Leskovec. Hyperbolic graph convolutional neural networks. In *Advances in neural information processing systems*, volume 32, 2019.

Yixin Chen, Jinbo Bi, and James Ze Wang. Miles: Multiple-instance learning via embedded instance selection. *IEEE transactions on pattern analysis and machine intelligence*, 28(12):1931–1947, 2006.

Veronika Cheplygina, David MJ Tax, and Marco Loog. Dissimilarity-based ensembles for multiple instance learning. *IEEE transactions on neural networks and learning systems*, 27(6):1379–1391, 2015.

Christopher De Sa, Frederic Sala, Albert Gu, and Christopher Re. Representation tradeoffs for hyperbolic embeddings. In *International Conference on Machine Learning*, 2018.

Mehmet Demircigil, Judith Heusel, Matthias Löwe, Sebastian Upgang, and Franck Vermet. On a model of associative memory with huge storage capacity. *Journal of Statistical Physics*, 168: 288–299, 2017.

Thomas G Dietterich, Richard H Lathrop, and Tomás Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. *Artificial intelligence*, 89(1-2):31–71, 1997.

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

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

Mikhael Gromov. Hyperbolic groups. In *Essays in group theory*, pp. 75–263. Springer, 1987.

Caglar Gülçehre, Misha Denil, Mateusz Malinowski, Ali Razavi, Razvan Pascanu, Karl Moritz Hermann, Phil Blunsom, and Nando de Freitas. Hyperbolic attention networks. In *International Conference on Learning Representations (ICLR)*, 2019.

Caglar Gulcehre, Anirudh Dey, Md Rahman, Misha Denil, and Yoshua Bengio. Hyperbolic attention networks. In *International Conference on Learning Representations*, 2019.

Benjamin Hoover, Yuchen Liang, Bao Pham, Rameswar Panda, Hendrik Strobelt, Duen Horng Chau, Mohammed Zaki, and Dmitry Krotov. Energy transformer. *Advances in neural information processing systems*, 36:27532–27559, 2023.

John J Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8):2554–2558, 1982.

Maximilian Ilse, Jakub Tomczak, and Max Welling. Attention-based deep multiple instance learning. In *International conference on machine learning*, pp. 2127–2136. PMLR, 2018.

Dejun Jiang, Zhenxing Wu, Chang-Yu Hsieh, Guangyong Chen, Ben Liao, Zhe Wang, Chao Shen, Dongsheng Cao, Jian Wu, and Tingjun Hou. Could graph neural networks learn better molecular representation for drug discovery? a comparison study of descriptor-based and graph-based models. *Journal of cheminformatics*, 13:1–23, 2021.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

Dmitri Krioukov, Fragkiskos Papadopoulos, Maksim Kitsak, Amin Vahdat, and Marián Boguná. Hyperbolic geometry of complex networks. *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics*, 82(3):036106, 2010.

Dmitry Krotov and John J Hopfield. Dense associative memory for pattern recognition. *Advances in neural information processing systems*, 29, 2016.

Emel Şeyma Küçükaşcı and Mustafa Gökçe Baydoğan. Bag encoding strategies in multiple instance learning problems. *Information Sciences*, 467:559–578, 2018.

Michael Kuhn, Ivica Letunic, Lars Juhl Jensen, and Peer Bork. The sider database of drugs and side effects. *Nucleic acids research*, 44(D1):D1075–D1079, 2016.

Nathan Linial, Eran London, and Yuri Rabinovich. The geometry of graphs and some of its algorithmic applications. *Combinatorica*, 15:215–245, 1995.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Ines Filipa Martins, Ana L Teixeira, Luis Pinheiro, and Andre O Falcao. A bayesian approach to in silico blood-brain barrier penetration modeling. *Journal of chemical information and modeling*, 52(6):1686–1697, 2012.

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

Ilia Ovinnikov, Oleg Rebane, Richard Socher, and Caiming Xiong. Hyperbolic variational autoencoders. In *International Conference on Learning Representations*, 2021.

Hubert Ramsauer, Bernhard Schäfl, Johannes Lehner, Philipp Seidl, Michael Widrich, Thomas Adler, Lukas Gruber, Markus Holzleitner, Milena Pavlović, Geir Kjetil Sandve, et al. Hopfield networks is all you need. *arXiv preprint arXiv:2008.02217*, 2020.

Hubert Ramsauer, Bernhard Sch"afl, Johannes Lehner, Philipp Seidl, Michael Widrich, Lukas Gruber, Markus Holzleitner, Michal Pavlovi'c, Geir Kjetil Sandve, Victor Greiff, et al. Hopfield networks is all you need. *Advances in Neural Information Processing Systems*, 34:9377–9391, 2021.

Frederic Sala, Christopher De Sa, Albert Gu, and Christopher Re. Representation tradeoffs for hyperbolic embeddings. In *International Conference on Machine Learning*, 2018.

Satoshi Shimizu, Daisuke Toyama, and Yutaka Miyake. Hyperbolic neural networks++: A hyperbolic embedding method for graph neural networks. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 2021.

Govindan Subramanian, Bharath Ramsundar, Vijay Pande, and Rajiah Aldrin Denny. Computational modeling of β-secretase 1 (bace-1) inhibitors using ligand based approaches. *Journal of chemical information and modeling*, 56(10):1936–1949, 2016.

David W Tank and John J Hopfield. Simple neural optimization networks: An a/d converter, signal decision circuit, and a linear programming circuit. *IEEE Transactions on Circuits and Systems*, 33(5):533–541, 1986.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, volume 30, 2017.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

Jun Wang and Jean-Daniel Zucker. Solving multiple-instance problem: A lazy learning approach. 2000.

Michael Widrich, Bernhard Schäfl, Milena Pavlović, Hubert Ramsauer, Lukas Gruber, Markus Holzleitner, Johannes Brandstetter, Geir Kjetil Sandve, Victor Greiff, Sepp Hochreiter, et al. Modern hopfield networks and attention for immune repertoire classification. *Advances in neural information processing systems*, 33:18832–18845, 2020.

Zhenqin Wu, Bharath Ramsundar, Evan N Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S Pappu, Karl Leswing, and Vijay Pande. Moleculenet: a benchmark for molecular machine learning. *Chemical science*, 9(2):513–530, 2018.

Zhaoping Xiong, Dingyan Wang, Xiaohong Liu, Feisheng Zhong, Xiaozhe Wan, Xutong Li, Zhaojun Li, Xiaomin Luo, Kaixian Chen, Hualiang Jiang, et al. Pushing the boundaries of molecular representation for drug discovery with the graph attention mechanism. *Journal of medicinal chemistry*, 63(16):8749–8760, 2019.

Alan L Yuille and Anand Rangarajan. The concave-convex procedure (cccp). *Advances in neural information processing systems*, 14, 2001.

A Hyperbolic Energy-based Optimization Framework

A.1 Bounding the Energy Function

We consider the energy

$$E(\xi) \;=\; -\frac{1}{\theta}\log\left(\sum_{i=1}^{S}\exp\big(\theta\,\langle x_i,\xi\rangle_M\big)\right) \;+\; \frac{1}{2}\,d_{\mathcal{M}}(\xi,p)^2, \tag{13}$$

where the (hyperbolic) similarity is $\langle x,y\rangle_M := -\cosh\big(d_{\mathcal{M}}(x,y)\big)$ on a complete, simply connected Riemannian manifold $(\mathcal{M},g)$ of constant negative curvature $-c$.

Setup and notation. Fix a base point $p\in\mathcal{M}$ and define

$$r_i := d_{\mathcal{M}}(x_i,p), \qquad r := d_{\mathcal{M}}(\xi,p).$$

Let $M_r := \max_i r_i$. We assume optimization is restricted (by standard clipping/projection) to a geodesic ball around p, i.e., $r\le R_r$. Here S is the number of stored patterns and $\theta>0$ the inverse temperature.

A.1.1 BOUNDING THE SIMILARITY

By the triangle inequality,

$$|r_i - r| \;\le\; d_{\mathcal{M}}(x_i, \xi) \;\le\; r_i + r.$$

Since $\cosh(\cdot)$ is strictly increasing on $[0, \infty)$ and $\langle x_i, \xi \rangle_M = -\cosh d_{\mathcal{M}}(x_i, \xi)$, we obtain for each i

$$-\cosh(r_i + r) \;\le\; \langle x_i, \xi \rangle_M \;\le\; -\cosh(|r_i - r|). \tag{14}$$

Consequently, using $r_i \le M_r$ and $r \le R_r$,

$$\boxed{-\cosh(M_r + R_r) \;\le\; \langle x_i, \xi \rangle_M \;\le\; -1} \qquad (\forall i), \tag{15}$$

because $\cosh(0) = 1$ and $|r_i - r|$ can be as small as 0.

A.1.2 BOUNDING THE ENERGY

Write $E(\xi) = E_{\text{cave}}(\xi) + E_{\text{cvx}}(\xi)$ with

$$E_{\text{cave}}(\xi) = -\frac{1}{\theta} \log \sum_{i=1}^{S} e^{\theta z_i}, \quad z_i := \langle x_i, \xi \rangle_M, \qquad E_{\text{cvx}}(\xi) = \tfrac{1}{2}\, d_{\mathcal{M}}(\xi, p)^2.$$

For any $\theta > 0$, the log-sum-exp bounds yield

$$\max_i z_i \;\le\; \frac{1}{\theta} \log \sum_i e^{\theta z_i} \;\le\; \max_i z_i + \frac{\log S}{\theta} \quad \Longrightarrow \quad -\max_i z_i - \frac{\log S}{\theta} \;\le\; E_{\text{cave}}(\xi) \;\le\; -\max_i z_i.$$

From equation 15 we have $\max_i z_i \in \big[-\cosh(M_r + R_r),\; -1\big]$. Hence

$$\boxed{1 - \frac{\log S}{\theta} \;\le\; E_{\text{cave}}(\xi) \;\le\; \cosh(M_r + R_r)}. \tag{16}$$

For the convex part (squared distance to p),

$$0 \;\le\; E_{\text{cvx}}(\xi) = \tfrac{1}{2}\, d_{\mathcal{M}}(\xi, p)^2 \;\le\; \tfrac{1}{2}\, R_r^2. \tag{17}$$

A.1.3 FINAL BOUNDS

Combining equation 16 and equation 17 yields

$$\boxed{1 - \frac{\log S}{\theta} \;\le\; E(\xi) \;\le\; \cosh(M_r + R_r) \;+\; \tfrac{1}{2}\, R_r^2}. \tag{18}$$

The constants depend only on the maximal radial extents of memories and states (M_r, R_r) and the inverse temperature θ, but *not* on the specific hyperbolic model. Thus the boundedness of E—and hence the numerical stability of CCCP or Riemannian-gradient iterations—holds uniformly across all constant-curvature hyperbolic realizations.

A.2 OPTIMIZATION OF THE ENERGY FUNCTION VIA CCCP

A.2.1 CONCAVITY/CONVEXITY ON HADAMARD MANIFOLDS

We work on a Hadamard manifold $(\mathcal{M}, g)$ with constant negative curvature. Write

$$E(\xi) \;=\; E_{\text{cvx}}(\xi) \;+\; E_{\text{cave}}(\xi), \qquad E_{\text{cvx}}(\xi) = \tfrac{1}{2}\, d_{\mathcal{M}}(\xi, p)^2, \quad E_{\text{cave}}(\xi) = -\tfrac{1}{\theta} \log \sum_{i=1}^{N} e^{\theta\, s_i(\xi)},$$

where $s_i(\xi) := \langle x_i, \xi \rangle_M = -\cosh(d_{\mathcal{M}}(x_i, \xi))$. It is known that $d_{\mathcal{M}}(\cdot, \cdot)$ is geodesically convex on Hadamard manifolds; since $\cosh$ is convex and strictly increasing on $[0, \infty)$, the composition $\cosh \circ d_{\mathcal{M}}$ is geodesically convex, hence $s_i(\xi) = -\cosh(d_{\mathcal{M}}(x_i, \xi))$ is *geodesically concave*. For the concavity of $E_{\text{cave}}(\xi) = -\operatorname{lse}_\theta(\{s_i(\xi)\}_i)$, let $F(\xi) = \operatorname{lse}_\theta(\{s_i(\xi)\}_i)$. For any unit tangent

vector $u \in T_\xi \mathcal{M}$, the (Riemannian) Hessian admits the standard decomposition (see the derivation in §A.3):

$$\mathrm{Hess}_\xi F[u,u] \;=\; \sum_{i=1}^{N} p_i(\xi)\,\mathrm{Hess}_\xi s_i[u,u] \;+\; \theta\,\mathrm{Var}_{p(\xi)}\big(\langle \mathrm{grad}\, s_i(\xi), u\rangle_g\big),$$

where $p_i(\xi) = \frac{e^{\theta s_i(\xi)}}{\sum_j e^{\theta s_j(\xi)}}$. Each s_i is geodesically concave, so $\mathrm{Hess}_\xi s_i[\cdot,\cdot] \preceq 0$; the second term is a nonnegative variance term. Therefore $\mathrm{Hess}_\xi(-F)[u,u] \;=\; -\sum_i p_i\,\mathrm{Hess}_\xi s_i[u,u] \;-\; \theta\,\mathrm{Var}_p(\cdot)$ is "a difference of a negative semidefinite and a positive semidefinite term." On a bounded geodesic ball, if there exists $\kappa > 0$ such that $-\mathrm{Hess}_\xi s_i \succeq \kappa I$ and $L := \max_{i,\xi} \|\mathrm{grad}\, s_i(\xi)\|_g < \infty$, then whenever $0 < \theta \le \kappa/L^2$ we have $\mathrm{Hess}_\xi(-F) \preceq 0$, hence E_{cave} is geodesically concave. Under this temperature range, $E = E_{\mathrm{cvx}} + E_{\mathrm{cave}}$ satisfies the "convex + concave" requirement for CCCP. In practice, we also observe monotone decrease under typical training temperatures.

A.2.2 CCCP Linearization and Surrogate

Let $\xi^{(t)}$ be the current iterate. A first-order (Riemannian) upper bound for the concave part yields

$$E_{\mathrm{cave}}(\xi) \;\le\; E_{\mathrm{cave}}(\xi^{(t)}) \;+\; \big\langle a^{(t)},\, \log_{\xi^{(t)}}(\xi)\big\rangle_{\xi^{(t)}}, \qquad a^{(t)} := \mathrm{grad}\, E_{\mathrm{cave}}(\xi^{(t)}).$$

Thus the "bound-minimization" surrogate for CCCP is

$$Q\Big(\xi \mid \xi^{(t)}\Big) \;=\; \tfrac{1}{2}\, d_{\mathcal{M}}(\xi,p)^2 \;+\; \big\langle a^{(t)},\, \log_{\xi^{(t)}}(\xi)\big\rangle_{\xi^{(t)}} \;+\; \text{const.} \tag{19}$$

A.2.3 Closed-Form Update (with Parallel Transport)

On a Hadamard manifold, the minimizer of equation 19 admits the closed form

$$\xi^{(t+1)} \;=\; \exp_p\Big(\; -\; \mathrm{PT}_{\xi^{(t)}\to p}\big(a^{(t)}\big)\Big), \tag{20}$$

where $\mathrm{PT}_{\xi^{(t)}\to p}$ denotes parallel transport along the unique geodesic from $\xi^{(t)}$ to p. For numerical stability, we employ a damped step with $\eta \in (0,1]$:

$$\xi^{(t+1)} \;=\; \exp_p\Big(\eta\, v^{(t)}\Big), \qquad v^{(t)} := -\,\mathrm{PT}_{\xi^{(t)}\to p}\big(a^{(t)}\big), \tag{21}$$

which reduces to equation 20 when $\eta = 1$.

A.2.4 Softmax Weights and Riemannian Gradient of the Concave Term

Let $p_i^{(t)} = \frac{\exp(\theta\, s_i(\xi^{(t)}))}{\sum_j \exp(\theta\, s_j(\xi^{(t)}))}$. By the chain rule together with §A.3, Eq. equation 24, we obtain

$$a^{(t)} = \mathrm{grad}\, E_{\mathrm{cave}}(\xi^{(t)}) = -\sum_{i=1}^{N} p_i^{(t)}\, \mathrm{grad}_\xi\, s_i(\xi)\Big|_{\xi=\xi^{(t)}}. \tag{22}$$

Convergence note When θ satisfies the above sufficient condition, E_{cvx} is geodesically convex and E_{cave} is geodesically concave; therefore CCCP guarantees that $E(\xi^{(t)})$ decreases monotonically and $\{\xi^{(t)}\}$ converges to a (meta-)stable memory attractor.

A.3 Riemannian Gradient of the Concave Term

Consider the hyperbolic similarity

$$s_M(x,y) := \langle x,y\rangle_M = -\cosh\big(d_{\mathcal{M}}(x,y)\big).$$

Let $d_{\mathcal{M}}$ denote the geodesic distance and $\log_x : \mathcal{M} \to T_x\mathcal{M}$ the Riemannian logarithm at x. On a Hadamard manifold, for any $x \neq y$,

$$\mathrm{grad}_x\, d_{\mathcal{M}}(x,y) \;=\; -\,\frac{\log_x(y)}{\|\log_x(y)\|_g}, \tag{23}$$

where $\|\cdot\|_g$ is the norm induced by g on $T_x\mathcal{M}$.

Chain rule for the similarity gradient By the chain rule,

$$\operatorname{grad}_x s_M(x,y) = -\sinh\big(d_{\mathcal{M}}(x,y)\big)\,\operatorname{grad}_x d_{\mathcal{M}}(x,y),$$

and substituting equation 23 yields

$$\boxed{\operatorname{grad}_x s_M(x,y) \;=\; \sinh\big(d_{\mathcal{M}}(x,y)\big)\,\frac{\log_x(y)}{\big\|\log_x(y)\big\|_g}\,.} \tag{24}$$

The gradient points along the unit tangent from x to y with magnitude $\sinh(d_{\mathcal{M}}(x,y))$.

Riemannian gradient of E_{cave} Let $s_i(\xi) = \langle x_i, \xi\rangle_M$ and $p_i(\xi) = \frac{e^{\theta s_i(\xi)}}{\sum_j e^{\theta s_j(\xi)}}$. Then

$$\boxed{\operatorname{grad} E_{\text{cave}}(\xi) = -\sum_{i=1}^{N} p_i(\xi)\,\operatorname{grad}_\xi s_i(\xi) \;=\; -\sum_{i=1}^{N} p_i(\xi)\,\sinh\big(d_{\mathcal{M}}(x_i,\xi)\big)\,\frac{\log_\xi(x_i)}{\big\|\log_\xi(x_i)\big\|_g}.} \tag{25}$$

Coordinate gradient (Poincaré ball example) If the chosen coordinates are conformal (e.g., the Poincaré ball), then $g(\xi) = \lambda(\xi)^2 I$ with $\lambda(\xi) = \frac{2}{1-c\|\xi\|^2}$. The Euclidean (coordinate) gradient ∇_ξ and the Riemannian gradient grad_ξ satisfy

$$\nabla_\xi f \;=\; G(\xi)^{-1}\operatorname{grad}_\xi f \;=\; \lambda(\xi)^{-2}\operatorname{grad}_\xi f. \tag{26}$$

Plugging equation 25 into equation 26 yields an implementation-ready Euclidean gradient expression.

B Supplementary Notes on Storage Capacity

We analyze the storage capacity of HAMNs on a Hadamard manifold $(\mathcal{M}, g)$ of constant negative curvature $-c < 0$. The similarity is $\langle x, y\rangle_M := -\cosh\big(d_{\mathcal{M}}(x,y)\big)$, and the (intrinsically regularized) energy is

$$E(\xi) \;=\; -\tfrac{1}{\theta}\log\sum_{i=1}^{N} e^{\theta\langle x_i,\xi\rangle_M} \;+\; \tfrac{1}{2}\,d_{\mathcal{M}}(\xi,p)^2,$$

where $p \in \mathcal{M}$ is a fixed reference point (see Sec. 3.2).

B.1 Energy-well separation and recallability

Let the stored patterns be $\{x_i\}_{i=1}^N \subset \mathcal{M}$, and define the minimum pairwise geodesic separation

$$\delta \;:=\; \min_{i\neq j} d_{\mathcal{M}}(x_i,x_j).$$

Fix any radius $\rho < \delta/2$. If the query lies in the intrinsic ball $\xi \in B_{\mathcal{M}}(x_k,\rho)$, then by the triangle inequality $d_{\mathcal{M}}(\xi,x_k) \le \rho$ and, for any $j \neq k$, $d_{\mathcal{M}}(\xi,x_j) \ge \delta - \rho \ge \delta/2 + \varepsilon$ with $\varepsilon := \delta/2 - \rho > 0$. Since $\cosh$ is increasing and $s_i(\xi) := \langle x_i,\xi\rangle_M = -\cosh d_{\mathcal{M}}(x_i,\xi)$, we obtain

$$s_k(\xi) \;\ge\; -\cosh(\rho), \qquad s_j(\xi) \;\le\; -\cosh(\delta-\rho) \quad (j\neq k),$$

hence the gap $s_k(\xi) - s_j(\xi) \;\ge\; \Delta(\delta,\varepsilon) := \cosh(\frac{\delta}{2}+\varepsilon) - \cosh(\frac{\delta}{2}-\varepsilon) = 2\sinh(\frac{\delta}{2})\sinh(\varepsilon) \;>\; 0$. This yields the softmax dominance bound

$$p_k(\xi) := \frac{\exp\big(\theta\, s_k(\xi)\big)}{\sum_j \exp\big(\theta\, s_j(\xi)\big)} \;\ge\; \frac{1}{1+(N-1)\exp\big(-\theta\,\Delta(\delta,\varepsilon)\big)}.$$

For sufficiently separated patterns (large δ) and/or a sharp energy (large θ), $p_k(\xi)$ is close to 1, and the Riemannian gradient of the concave term

$$a(\xi) = \operatorname{grad} E_{\text{cave}}(\xi) = -\sum_{i=1}^{N} p_i(\xi)\operatorname{grad}_\xi s_i(\xi) = -\sum_{i=1}^{N} p_i(\xi)\,\sinh\big(d_{\mathcal{M}}(x_i,\xi)\big)\,\frac{\log_\xi(x_i)}{\|\log_\xi(x_i)\|_g}$$

(see App. §A.3) is nearly aligned with the unit tangent toward x_k. More explicitly, choosing

$$\theta \;\geq\; \frac{1}{\Delta(\delta,\varepsilon)}\Big(\log\big(2(N-1)\big) \;+\; \log\tfrac{\sinh(\delta-\rho)}{\sinh(\rho)}\Big),$$

ensures $\langle -a(\xi),\, u_k\rangle_g \geq 0$ (u_k the unit tangent from ξ to x_k), so the CCCP update $\xi^+ = \exp_p\big(-\mathrm{PT}_{\xi\to p}(a(\xi))\big)$ moves ξ toward x_k and keeps it within $B_{\mathcal{M}}(x_k,\rho)$. Under this condition, each x_k induces a stable well and an attraction basin.

Effect of the intrinsic regularizer The intrinsic penalty $\frac{1}{2}d_{\mathcal{M}}(\xi,p)^2$ suppresses excursions far from p. The dominance bound for $p_k(\xi)$ is determined solely by the concave term and is unaffected by the regularizer; its role appears in the CCCP closed-form step taken at p, which improves numerical stability and step-size control.

B.2 Hyperbolic volume and a sphere-packing upper bound

Assume all patterns lie in a geodesic ball $B_{\mathcal{M}}(p,R)$. If the recall basins $B_{\mathcal{M}}(x_i,\delta/2)$ are pairwise disjoint, then

$$N_{\max} \;\leq\; \frac{\mathrm{Vol}\big(B_{\mathcal{M}}(p,R)\big)}{\mathrm{Vol}\big(B_{\mathcal{M}}(\cdot,\delta/2)\big)}.$$

In a d-dimensional hyperbolic space of curvature $-c$, the ball volume satisfies

$$\mathrm{Vol}_c\big(B(r)\big) = \omega_{d-1}\int_0^r \Big(\tfrac{\sinh(\sqrt{c}\,t)}{\sqrt{c}}\Big)^{d-1} dt \;\asymp\; \kappa_{d,c}\,\exp\big((d-1)\sqrt{c}\,r\big) \quad (r \gg 1/\sqrt{c}),$$

whence

$$\boxed{N_{\max} \;\lesssim\; \exp\Big((d-1)\sqrt{c}\,\big(R-\tfrac{\delta}{2}\big)\Big) \;=\; \exp\big(\alpha_{\mathrm{hyp}}\,(R-\tfrac{\delta}{2})\big), \quad \alpha_{\mathrm{hyp}} := (d-1)\sqrt{c}.}$$

Thus capacity grows *exponentially* in the radius with a rate controlled by both the dimension d and curvature c: very small c under-expresses hierarchy (small rate), whereas overly large c increases metric distortion and may hurt optimization.

B.3 Comparison with Euclidean MHNs

Modern Euclidean Hopfield networks can achieve exponential capacity in the ambient dimension for random patterns (e.g., $N = 2^{\Omega(d)}$ under log-sum-exp energy). Our hyperbolic packing bound is complementary: due to the *exponential volume growth* of negatively curved spaces, when hierarchical data concentrate outward along the radius (depth), the number of non-overlapping basins grows exponentially with $(d-1)\sqrt{c}$. This aligns with our empirical advantages on deep hierarchies.

Takeaway Error-free recall is ensured by a geometric margin δ; the total number of recallable patterns is upper-bounded by a hyperbolic sphere-packing law scaling as $\exp\big((d-1)\sqrt{c}\,(R-\delta/2)\big)$. This complements classical Euclidean capacity results and explains why HAMNs benefit more as hierarchical depth (effective hyperbolic radius) increases.

C Key Formulas for Common Hyperbolic Models (Constant Curvature $-c < 0$)

Notation & convention. Let the curvature be $-c$ with $c > 0$ and write $\mathrm{arcosh}(\cdot)$ for the inverse hyperbolic cosine. All standard hyperbolic models are *isometric*; hence any model-agnostic derivation in the paper becomes an implementation by choosing $d_{\mathcal{M}},\ \exp_p^c,\ \log_p^c$ from a specific model. We use the hyperbolic similarity

$$\langle x,y\rangle_M \;:=\; -\cosh\big(d_{\mathcal{M}}(x,y)\big),$$

with the universal chain rule

$$\nabla_x\langle x,y\rangle_M = -\,\sinh\big(d_{\mathcal{M}}(x,y)\big)\;\nabla_x d_{\mathcal{M}}(x,y),$$

equivalently $\mathrm{grad}_x\langle x,y\rangle_M = \sinh(d_{\mathcal{M}}(x,y))\,\frac{\log_x(y)}{\|\log_x(y)\|_g}$ (see App. §A.3).

Model-agnostic CCCP closed form (used in this work). Let

$$a(\xi) \;:=\; \operatorname{grad} E_{\text{cave}}(\xi) \;=\; -\sum_{i=1}^{N} p_i(\xi)\, \operatorname{grad}_\xi \langle x_i, \xi\rangle_M, \qquad p_i(\xi) = \frac{e^{\theta\langle x_i,\xi\rangle_M}}{\sum_j e^{\theta\langle x_j,\xi\rangle_M}}.$$

Our CCCP step reads

$$v \;:=\; -\operatorname{PT}_{\xi\to p}\big(a(\xi)\big), \qquad \xi^+ \;=\; \exp_p^c\big(\eta\, v\big),\;\; \eta\in(0,1]. \tag{27}$$

Thus each model only needs $\exp_p^c, \log_p^c, d_{\mathcal{M}}$ (and, if desired, a convenient form of parallel transport).

C.1 Poincaré Ball $\mathbb{D}_c^d = \{x \in \mathbb{R}^d : \|x\| < 1/\sqrt{c}\}$

Exponential/log at the origin

$$\exp_0^c(v) = \tanh(\sqrt{c}\,\|v\|)\,\frac{v}{\sqrt{c}\,\|v\|}, \qquad \log_0^c(x) = \operatorname{artanh}(\sqrt{c}\,\|x\|)\,\frac{x}{\sqrt{c}\,\|x\|}.$$

Base-point maps (with Möbius addition $\oplus_c$ and conformal factor $\lambda_x^c = \frac{2}{1-c\|x\|^2}$)

$$\exp_p^c(v) = p\oplus_c\Big(\tanh\big(\tfrac{\sqrt{c}\,\lambda_p^c\|v\|}{2}\big)\,\tfrac{v}{\sqrt{c}\,\|v\|}\Big), \quad \log_p^c(x) = \frac{2}{\sqrt{c}\,\lambda_p^c}\,\operatorname{artanh}\big(\sqrt{c}\,\|(-p)\oplus_c x\|\big)\,\frac{(-p)\oplus_c x}{\|(-p)\oplus_c x\|}.$$

Geodesic distance

$$d_{\mathbb{D}_c}(x,y) = \frac{1}{\sqrt{c}}\,\operatorname{arcosh}\left(1 + \frac{2c\|x-y\|^2}{(1-c\|x\|^2)(1-c\|y\|^2)}\right).$$

Parallel transport & implementation note. The ball is *conformal*, $g(x) = \lambda(x)^2 I$ with $\lambda(x) = \frac{2}{1-c\|x\|^2}$. Choosing $p = 0$, the transport along the unique geodesic to the origin can be implemented as a scalar rescaling $\operatorname{PT}_{\xi\to 0}(u) = \frac{\lambda(\xi)}{\lambda(0)}u$. This scaling preserves the Riemannian norm because the ball is conformal: letting $\|\cdot\|_E$ denote the Euclidean norm, $\|u'\|_E = \frac{\lambda(\xi)}{\lambda(0)}\|u\|_E$ ensures $\lambda(0)^2\|u'\|_E^2 = \lambda(\xi)^2\|u\|_E^2$. Then equation 27 amounts to $v_0 := -(\lambda(\xi)/\lambda(0))\,a(\xi)$ followed by $\exp_0^c$.

Möbius addition / scalar multiplication.

$$u\oplus_c v = \frac{\big(1 + 2c\langle u,v\rangle + c\|v\|^2\big)u + (1-c\|u\|^2)v}{1 + 2c\langle u,v\rangle + c^2\|u\|^2\|v\|^2}, \qquad r\otimes_c u = \frac{1}{\sqrt{c}}\tanh\big(r\,\operatorname{artanh}(\sqrt{c}\,\|u\|)\big)\,\frac{u}{\|u\|}.$$

C.2 Upper Half-Plane $\mathbb{H}_c = \{(u,y) \in \mathbb{R}^{d-1}\times\mathbb{R}_{>0}\}$

Exponential/log at $o = (0,\dots,0,1/\sqrt{c})$ Let $v = (v_u, v_y) \in \mathbb{R}^{d-1}\times\mathbb{R}$. Then

$$\exp_o^c(v) = \Big(e^{\sqrt{c}\,v_y}\,v_u,\;\; \tfrac{1}{\sqrt{c}}\,e^{\sqrt{c}\,v_y}\Big), \qquad \log_o^c(u,y) = \frac{1}{\sqrt{c}}\Big(\frac{u}{y},\; \ln(\sqrt{c}\,y)\Big).$$

Geodesic distance

$$d_{\mathbb{H}_c}\big((u_1,y_1),(u_2,y_2)\big) = \frac{1}{\sqrt{c}}\,\operatorname{arcosh}\left(1 + \frac{c\big(\|u_1-u_2\|^2 + (y_1-y_2)^2\big)}{2y_1y_2}\right).$$

Parallel transport & implementation note. This model is conformal with $g(u,y) = \lambda(u,y)^2 I,\ \lambda(u,y) = \frac{1}{\sqrt{c}\,y}$. If $p = o$, then $\operatorname{PT}_{\xi\to o}(w) = \frac{\lambda(\xi)}{\lambda(o)}w = \frac{1}{\sqrt{c}\,y_\xi}\,w$, followed by $\exp_o^c$.

C.3 Klein Model $\mathbb{K}_c^d = \{x \in \mathbb{R}^d : \|x\| < 1/\sqrt{c}\}$

Exponential/log at the origin coincide with the ball at 0: $\exp_0^c,\ \log_0^c$ as above. **Geodesic distance**

$$d_{\mathbb{K}_c}(p,q) = \operatorname{arcosh}\left(\frac{1 - c\,p^\top q}{\sqrt{(1-c\|p\|^2)(1-c\|q\|^2)}}\right).$$

Parallel transport: recommendation. Since the Klein model is not conformal, closed-form PT is more involved. In practice, use an isometry to Lorentz (or Poincaré), perform PT and the exponential step there, and map back to Klein.

C.4 HEMISPHERE $J_c = \{u \in \mathbb{S}^n : u_{n+1} > 0\}$

Implementation note. We recommend using the standard isometry to the Lorentz (hyperboloid) model to compute $\exp / \log$, d, and PT, and then map back to the hemisphere. We omit redundant explicit formulas here to avoid confusion, since our experiments instantiate Poincaré/Lorentz directly.

C.5 LORENTZ (HYPERBOLOID) $L_c = \{X \in \mathbb{R}^{n+1} : X_0^2 - \sum_{i=1}^n X_i^2 = \frac{1}{c},\ X_0 > 0\}$

Minkowski bilinear form. $(X, Y)_M = -X_0 Y_0 + \sum_{i=1}^n X_i Y_i$.

Distance and similarity.

$$d_{L_c}(X, Y) = \frac{1}{\sqrt{c}} \operatorname{arcosh}\big(-c\,(X, Y)_M\big), \qquad \langle X, Y\rangle_M = -\cosh\big(d_{L_c}(X, Y)\big).$$

Exponential/logarithm at $e_0 = (\frac{1}{\sqrt{c}}, 0, \ldots, 0)$.

$$\exp^c_{e_0}(W) = \Big(\tfrac{1}{\sqrt{c}} \cosh(\sqrt{c}\,\|W\|),\ \tfrac{1}{\sqrt{c}} \sinh(\sqrt{c}\,\|W\|) \tfrac{W}{\|W\|}\Big), \quad \log^c_{e_0}(X) = \frac{1}{\sqrt{c}} \operatorname{arcosh}(\sqrt{c}\,X_0) \frac{X_{1:n}}{\|X_{1:n}\|}.$$

Parallel transport. It can be implemented by a Lorentz boost: let $B_{X \to e_0} \in SO^+(1, n)$ map X to e_0, then $\mathrm{PT}_{X \to e_0}(V) = B_{X \to e_0} V$.

Where to plug in the main text.

- **Memory encoding (Sec. 3, Eq. equation 4).** Pick $\exp^c_p, \log^c_p$ from any model. If tangent clipping is used, clip in $T_p\mathcal{M}$ and map back via $\exp^c_p$.
- **Energy (Eq. equation 6).** Substitute the chosen model's $d_{\mathcal{M}}$ (or equivalently $\langle \cdot, \cdot \rangle_M$); nothing else changes.
- **Retrieval/optimization (CCCP step equation 27).** Use the model-agnostic gradient of the concave term via $\log_\xi(\cdot)$ and $\sinh d$, then apply the model's PT and $\exp^c_p$ to update.
- **Energy bounds (Appendix A.1).** Plug the model's $d_{\mathcal{M}}$ into the same bounding argument.

D HYPERBOLIC HOPFIELD LAYERS FOR DEEP LEARNING

To seamlessly integrate hyperbolic associative memory into end-to-end networks, we construct three core modules on the Poincaré ball $\mathbb{D}^d_c$:

Hyperbolic Hopfield, **Hyperbolic Hopfield Pooling**, **Hyperbolic Hopfield Layer**.

All three share the curvature parameter c (which can be made learnable) and follow the CCCP closed-form step derived in the main text: first parallel-transport the Riemannian gradient of the concave term to a reference point p, then take the exponential-map update at p; in our implementation we set $p = 0$ (the ball center).

D.1 HYPERBOLIC HOPFIELD

HypHopfield takes queries $R \in \mathbb{R}^{B \times d}$ and memories $Y \in \mathbb{R}^{N \times d}$ as input, and outputs $Z \in \mathbb{R}^{B \times d}$. It implements the retrieval update on $\mathbb{D}^d_c$ (see Appendix §C).

1. **Hyperbolic attention (similarity and soft weights)**

$$S_{b,i} = \langle Y_i, R_b\rangle_M = -\cosh\big(d_{\mathbb{D}_c}(Y_i, R_b)\big), \qquad P_{b,i} = \frac{e^{\theta S_{b,i}}}{\sum_{j=1}^N e^{\theta S_{b,j}}},$$

yielding $P \in \mathbb{R}^{B \times N}$.

Algorithm 1 HypHopfield retrieval on the Poincaré ball $\mathbb{D}_c^d$

Require: Memories $Y = \{Y_i\}_{i=1}^N \subset \mathbb{D}_c^d$, queries $R^{(0)} = \{R_b^{(0)}\}_{b=1}^B \subset \mathbb{D}_c^d$, curvature $c > 0$, temperature $\theta > 0$, stepsize $\eta \in (0, 1]$, base point $p = 0$, max iters $T_{\max}$, tolerance ε

1: **for** $t = 0, 1, \ldots, T_{\max} - 1$ **do**

2: **Hyperbolic similarities:** $S_{b,i} \leftarrow -\cosh\big(d_{\mathbb{D}_c}(Y_i, R_b^{(t)})\big)$

3: **Soft weights:** $P_{b,i} \leftarrow \exp(\theta S_{b,i}) \big/ \sum_j \exp(\theta S_{b,j})$

4: **Riemannian gradient of concave term at $R_b^{(t)}$:**

$$a_b \leftarrow -\sum_{i=1}^{N} P_{b,i}\ \sinh\big(d_{\mathbb{D}_c}(Y_i, R_b^{(t)})\big)\ \frac{\log_{R_b^{(t)}}(Y_i)}{\|\log_{R_b^{(t)}}(Y_i)\|_g}$$

5: **Parallel transport to $p = 0$:** $v_b \leftarrow -\operatorname{PT}_{R_b^{(t)} \to 0}(a_b)$

6: **Base-point update (CCCP with damping):** $R_b^{(t+1)} \leftarrow \exp_0^c\big(\eta\, v_b\big)$, project back to $\mathbb{D}_c^d$ if needed

7: **Stopping:** **if** $d_{\mathbb{D}_c}(R_b^{(t+1)}, R_b^{(t)}) < \varepsilon$ for all b **then break**

8: **end for**

9: **Output:** $Z = \{R_b^{(t+1)}\}_{b=1}^B$

2. **Intrinsic gradient and parallel transport to the base point** Let the concave term be $E_{\text{cave}}(\xi) = -\frac{1}{\theta} \log \sum_i e^{\theta \langle x_i, \xi \rangle_M}$. For each batch element R_b, the *Riemannian gradient* of the concave term is (see Appendix §A.3)

$$a_b \;=\; \operatorname{grad} E_{\text{cave}}(R_b) \;=\; -\sum_{i=1}^{N} P_{b,i}\, \operatorname{grad}_\xi \langle Y_i, \xi \rangle_M \Big|_{\xi = R_b},$$

where $\operatorname{grad}_\xi \langle Y_i, \xi \rangle_M = \sinh\big(d_{\mathbb{D}_c}(Y_i, \xi)\big)\, \frac{\log_\xi(Y_i)}{\|\log_\xi(Y_i)\|_g}$. Parallel-transport this direction to the reference point $p = 0$:

$$v_b \;:=\; -\operatorname{PT}_{R_b \to 0}\big(a_b\big).$$

In the *conformal* Poincaré model, the metric is $g(x) = \lambda(x)^2 I$ with $\lambda(x) = \frac{2}{1 - c\|x\|^2}$. We adopt the transport rule consistent with our implementation (see Appendix §C):

$$\operatorname{PT}_{R_b \to 0}(u) \;=\; \frac{\lambda(R_b)}{2}\, u, \quad \Rightarrow \quad v_b \;=\; -\frac{\lambda(R_b)}{2}\, a_b.$$

(Using the exact PT of the model is also possible; empirically the results are consistent with our implementation.)

3. **Base-point exponential map (at $p = 0$)** Update with stepsize $\eta \in (0, 1]$:

$$Z_b \;=\; \exp_0^c\big(\,\eta\, v_b\,\big) \;=\; \tanh\big(\sqrt{c}\,\|\eta v_b\|\big)\, \frac{\eta v_b}{\sqrt{c}\,\|\eta v_b\|},$$

and apply ball projection when necessary to avoid numerical issues near the boundary (standard clipping in our implementation).

Implementation hints (consistent with code) (1) If upstream features are in Euclidean coordinates, first map them to the ball with $\mathrm{ToPoincaré}$ and then perform the three steps above; if Euclidean outputs are required, apply $\mathrm{FromPoincaré}$ at the end. (2) All submodules in this paper share the same curvature handle c (either learnable or fixed).

Pseudocode. The retrieval step is summarized in Alg. 1.[2]

[2]For parallel transport (PT) we use the exact PT (via an isometry to the Lorentz model); our code also provides the conformal-scaling approximation $\operatorname{PT}_{x \to 0}(u) = \frac{\lambda(x)}{2} u$ with $\lambda(x) = \frac{2}{1 - c\|x\|^2}$, which yielded similar results in our experiments.

Table 3: Results for MIL datasets *Tiger*, *Fox*, *Elephant* (AUC). Except for our method, results are from Ramsauer et al. (2020).

Method	tiger	fox	elephant
HAMNs(ours)	89.0 ± 0.4	$\mathbf{77.3 \pm 0.8}$	92.8 ± 0.2
MHNsRamsauer et al. (2020)	$\mathbf{91.3 \pm 0.5}$	64.05 ± 0.4	$\mathbf{94.9 \pm 0.3}$
Path encoding Küçükaşcı & Baydoğan (2018)	91.0 ± 1.0	71.2 ± 1.4	94.4 ± 0.7
MInD Cheplygina et al. (2015)	85.3 ± 1.1	70.4 ± 1.6	93.6 ± 0.9
MILES Chen et al. (2006)	87.2 ± 1.7	73.8 ± 1.6	92.7 ± 0.7
APR Dietterich et al. (1997)	77.8 ± 0.8	54.1 ± 0.9	55.0 ± 1.0
Citation-kNN Wang & Zucker (2000)	85.5 ± 0.9	63.5 ± 1.5	89.6 ± 0.9

D.2 Hyperbolic Hopfield Pooling

HypPooling aggregates m learnable *query* vectors $R \in \mathbb{R}^{m \times d}$ and N instance embeddings (as *memories*) $Y \in \mathbb{R}^{N \times d}$ into m hyperbolic summary vectors. Its computation is identical to HypHopfield (hyperbolic attention $\rightarrow$ Riemannian gradient with PT to $p = 0 \rightarrow$ base-point exponential map), except that R is a fixed-size learnable parameter while Y comes from upstream instances or outputs of previous layers. We validate its effectiveness in multi-instance learning tasks.

D.3 Hyperbolic Hopfield Layer

HypLayer propagates a small number of queries (input vectors) R through a learnable *memory matrix* $Y \in \mathbb{R}^{N \times d}$; Y can be initialized from a reference set (class prototypes, training-set embeddings, etc.) and trained. The update rule is the same as HypHopfield (base-point exponential update at $p = 0$), thereby supporting prototype/similarity-based retrieval, nearest-neighbor matching, and pattern aggregation; we verify its effectiveness in CIFAR-100 hierarchical classification and molecular property prediction tasks.

E Experiments

E.1 EXPERIMENT 1: MULTIPLE INSTANCE LEARNING DATASETS.

Table 4: Hyperparameter search space for manual selection on the Elephant, Fox, and Tiger validation sets.

Parameter	Values
Learning rates	$\{10^{-3}, 10^{-5}\}$
Learning rate decay (γ)	$\{0.98, 0.96, 0.94\}$
Number of heads	$\{8, 12, 16, 32\}$
Hidden dimensions	$\{32, 64, 128\}$
Bag dropout	$\{0.0, 0.75\}$
Poincaré curvature (c)	$\{1.0, 0.5, 0.1\}$
Clipping threshold ($\texttt{clip}_r$)	$\{0.9, 1.2, 2.8\}$
RSGD max iterations	$\{1, 5, 10\}$
RSGD learning rate (η)	$\{1.0, 0.1, 0.001\}$

To evaluate the performance of our Hyperbolic Associative Memory Networks (HAMNs) on multi–instance learning (MIL) tasks Dietterich et al. (1997), we conduct experiments on three classical benchmark datasets: **Tiger**, **Elephant**, and **Fox** (originally introduced by Ilse et al. (2018); Küçükaşcı & Baydoğan (2018); Carbonneau et al. (2018)). Each dataset consists of color images that are segmented into multiple regions and thus form a set of instances (segments or blobs); each instance is represented by color, texture, and shape descriptors. The learning objective is to classify the entire bag according to the presence of certain positive instances, despite the absence of instance–level annotations.

We introduce the proposed **HypPooling** module, which aggregates instance–level embeddings into a fixed–dimensional bag representation. Given a set of embedded instances as stored memory patterns Y (already mapped into hyperbolic space), we further introduce a set of *static and learnable query vectors* as state (query) patterns R, which also reside in the same Poincaré ball. Each query retrieves similar patterns from memory via a hyperbolic attention mechanism, thereby constructing a compressed representation of the input bag.

Elephant, Fox and Tiger are MIL datasets Andrews et al. (2002) for image annotation which comprise color images from the Corel dataset that have been preprocessed and segmented. An image consists of a set of segments (or blobs), each characterized by color, texture and shape descriptors. The datasets have 100 positive and 100 negative example images. The latter have been randomly drawn from a pool of photos of other animals. Elephant has 1391 instances and 230 features. Fox has 1320 instances and 230 features. Tiger has 1220 instances and 230 features. We used the HypPooling layer to perform hyperbolic aggregation of the input instances, and conducted a manual hyperparameter search on a validation set. Specifically, on the Elephant, Fox, and Tiger datasets we used the following architecture:

1. A fully connected linear embedding layer with ReLU activation;
2. Our **HypPooling** layer to perform the hyperbolic pooling operation on the embeddings;
3. A final ReLU–linear block as the classification output layer.

Results (Table 3) show that HAMNs match or outperform prior baselines and achieve the best score on **Fox**; overall, they remain comparable to Euclidean MHNs.

Among various hyperparameters, we focused particularly on those of the **HypPooling** layer, including the curvature c, the number of Riemannian gradient steps, and the learning rate η. All models were trained for 160 epochs using the AdamW optimizerLoshchilov & Hutter (2017) with exponential learning rate decay (see Table 4). We validated performance using 10-fold nested cross-validation repeated five times with different data splits; the reported ROC AUC scores are the averages across these runs. We also applied bag dropout at the bag level as our regularization technique.

E.2 EXPERIMENT 2: DRUG DESIGN BENCHMARK DATASETS.

To evaluate the effectiveness of our proposed Hyperbolic Associative Memory Networks (HAMNs) on molecular property prediction, we conduct experiments on four representative datasets from MoleculeNet (Wu et al., 2018). These datasets represent four main modeling tasks in drug design: (a) HIV for anti-viral activity prediction, introduced by the Drug Therapeutics Program (DTP) AIDS Antiviral Screen; (b) BACE for human β-secretase inhibitors (Subramanian et al., 2016); (c) BBBP for predicting blood-brain barrier permeability (Martins et al., 2012); and (d) SIDER for predicting drug side effects (Kuhn et al., 2016).

We apply the proposed **HypLayer** to the above molecular prediction tasks. Specifically, the training samples are used as stored memory patterns Y, while the input samples serve as state (query) patterns R. Each input is first mapped into the Poincaré ball via hyperbolic embedding, then undergoes state

Table 5: Results on drug design benchmark datasets. Predictive performance (ROCAUC) on test set as reported byJiang et al. (2021) for 50 random splits

Method	HIV	BACE	BBBP	SIDER
HAMNs(ours)	78.5 ± 2.6	87.2 ± 3.0	$\mathbf{90.2 \pm 2.5}$	$\mathbf{62.1 \pm 2.3}$
MHNs	79.3 ± 2.4	88.4 ± 1.5	89.1 ± 1.7	61.8 ± 2.6
Attentive FP	74.8 ± 1.5	70.8 ± 3.3	84.1 ± 2.2	56.2 ± 1.5
GCN	77.5 ± 1.6	63.2 ± 4.5	79.2 ± 3.9	55.4 ± 1.2
GAT	72.1 ± 3.6	77.4 ± 3.0	83.7 ± 2.0	56.4 ± 1.5
DNN	73.0 ± 1.8	86.5 ± 2.2	87.6 ± 2.0	62.0 ± 1.8
RF	$\mathbf{82.3 \pm 2.2}$	89.2 ± 1.2	90.0 ± 2.0	-
SVM	-	$\mathbf{89.3 \pm 1.5}$	89.4 ± 2.1	-

evolution through the Hopfield retrieval mechanism in hyperbolic space, and eventually converges to a stable point close to one of the memory patterns. The final prediction is determined based on the association between the converged state and the corresponding label in memory.

Table 6: Hyperparameter search-space for grid-search on HIV, BACE, BBBP and SIDER. All models were trained, if applicable, for 4 epochs using Adam and a batch size of 1 sample.

Parameter	**Values**
Learning rates	$\{0.0002\}$
Number of heads	$\{1, 32, 128, 512\}$
Dropout	$\{0.0, 0.1, 0.2\}$
Poincaré curvature (c)	$\{1.0, 0.5, 0.1\}$
Clipping threshold $(\texttt{clip}_r)$	$\{0.9, 1.2, 2.8\}$
RSGD max iterations	$\{1, 5, 10\}$
RSGD learning rate (η)	$\{1.0, 0.1, 0.001\}$
quantity	$\{2, 4, 8\}$

We compare HAMNs against several representative baselines, including Support Vector Machines (SVM), Random Forest (RF), Deep Neural Networks (DNN), and state-of-the-art graph neural networks: Graph Convolutional Networks (GCN) (Kipf & Welling, 2016), Graph Attention Networks (GAT) (Veličković et al., 2017), AttentiveFP (Xiong et al., 2019), and modern Hopfield networks (MHNs) (Ramsauer et al., 2020). All models follow the standard splitting protocol provided by MoleculeNet. We report the average AUC over 50 random splits for each dataset.

As shown in Table 5, our method achieves competitive performance across all datasets and sets a new state-of-the-art result on **BBBP**(AUC = 90.2 ± 2.5), **SIDER** (AUC = 62.1 ± 2.3). All hyperparameters were selected on separate validation sets and we selected the model with the highest validation AUC on five different random splits. (see Table 6)

E.3 EXPERIMENT 3: ABLATIONS: CURVATURE AND NUMBER OF STORED PATTERNS.

We also added a comparative experiment to study the effect of curvature c and the number of stored patterns:

Table 7: Comparison of curvature c (higher is better).

c	**flat_top**	**flat_super**	**flat_coarse**	**flat_fine**
0.1	0.8834	0.7501	0.5891	0.3524
0.2	0.8784	0.7366	0.5614	0.2798
0.3	0.8656	0.7225	0.5888	0.3498
0.4	0.8942	0.7629	0.6342	0.3757
0.5	0.8897	0.7439	0.6142	0.3617
0.6	0.8755	0.7687	0.6493	0.4372
0.7	0.9030	0.7774	0.6695	0.4823
0.8	0.8841	0.7571	0.6204	0.4505
0.9	0.8898	0.7675	0.6514	0.4563
1.0	0.8818	0.7592	0.6522	0.4715
2.0	0.8919	0.7570	0.6455	0.4737
3.0	0.8902	0.7624	0.6461	0.4467
4.0	0.8924	0.7711	0.6459	0.4112
5.0	0.8577	0.7541	0.6395	0.4099
6.0	0.8807	0.7429	0.5745	0.2680
7.0	0.8860	0.7521	0.6262	0.3536
8.0	0.8704	0.7446	0.6035	0.3288
9.0	0.8715	0.7280	0.5661	0.2162
10.0	0.8617	0.6864	0.4368	0.1468

The curvature c comparison data above come from hierarchical classification results on CIFAR-100 after imposing a four-level structured hierarchy.

Table 8: Comparison of number of stored patterns on CIFAR-100 (4-level hierarchy).

stored_n	flat_top	flat_super	flat_coarse	flat_fine
100	0.8898	0.7517	0.6419	0.4498
150	0.8893	0.7680	0.6563	0.4670
200	0.8826	0.7394	0.6413	0.4754
250	0.8710	0.7466	0.6494	0.4642
300	0.8917	0.7563	0.6463	0.4596
350	0.8889	0.7641	0.6431	0.4505
400	0.8954	0.7630	0.6379	0.4558
450	0.8945	0.7649	0.6282	0.4290
500	0.8851	0.7586	0.6458	0.4728
550	0.8928	0.7674	0.6400	0.4647
600	0.8932	0.7491	0.6426	0.4740
650	0.8916	0.7796	0.6507	0.4643
700	0.8893	0.7676	0.6513	0.4749
750	0.8932	0.7473	0.6419	0.4547
800	0.8836	0.7778	0.6575	0.4780

From this experiment we observe that choosing a *moderate* curvature (e.g. $c \approx 0.7$–2.0) yields the best trade-off across multiple hierarchy levels. Extreme curvatures should be avoided: too small a c fails to express the hierarchy well, while too large a c leads to degraded performance. (see Table 7)

Similarly, the stored-pattern comparison uses CIFAR-100 with a four-level hierarchy. Since CIFAR-100 contains 100 classes, the number of stored patterns starts at 100 and is increased for comparison.

From the table, accuracies at each level are not monotonic: too many stored patterns can degrade the top level, the middle level benefits from a somewhat larger pattern count, and the coarse/fine levels peak at high capacity (800). Overall the metrics fluctuate little across different pattern counts, indicating the model is fairly robust to this choice. Empirically, very large pattern counts increase computation cost, so using a somewhat smaller number incurs little performance loss while saving resources. (see Table 8)

THE USE OF LARGE LANGUAGE MODELS (LLMS)

We used large language models (LLMs) as a general-purpose assistant only for: (i) translation and grammar correction; (ii) text polishing and wording refinement; and (iii) suggesting intermediate steps or equivalent formulations in a small subset of mathematical derivations. Specifically, the LLM provided text-level assistance when drafting or rewriting the following parts: the model-agnostic gradient form of the hyperbolic similarity (Appendix A.3) and the structured presentation of upper/lower bounds for the energy function (Appendix A.1). All assumptions, derivations, and final proofs were independently re-derived, verified, and corrected by the authors as needed.