Hyperbolic vs Euclidean Embeddings in Few-Shot Learning: Two Sides of the Same Coin (WACV 2024)

presented by Choeun Kim

September 7, 2025

Seoul National University

Few-shot classification

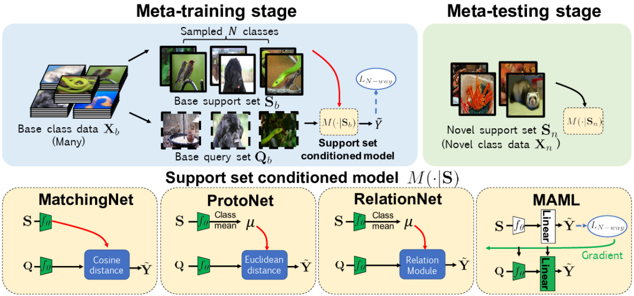


Figure 2: **Meta-learning few-shot classification algorithms.** The meta-learning classifier $M(\cdot|\mathbf{S})$ is conditioned on the support set $\mathbf{S}$. (*Top*) In the meta-train stage, the support set $\mathbf{S}_b$ and the query set $\mathbf{Q}_b$ are first sampled from random N classes, and then train the parameters in $M(.|\mathbf{S}_b)$ to minimize the N-way prediction loss $L_{\mathrm{N-way}}$. In the meta-testing stage, the adapted classifier $M(.|\mathbf{S}_n)$ can predict novel classes with the support set in the novel classes $\mathbf{S}_n$. (*Bottom*) The design of $M(\cdot|\mathbf{S})$ in different meta-learning algorithms.

Few-shot classification

Algorithm 1 Training episode loss computation for prototypical networks. N is the number of examples in the training set, K is the number of classes in the training set, $N_C \leq K$ is the number of classes per episode, N_S is the number of support examples per class, N_Q is the number of query examples per class. $\text{RANDOMSAMPLE}(S, N)$ denotes a set of N elements chosen uniformly at random from set S, without replacement.

Input: Training set $\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$, where each $y_i \in \{1, \ldots, K\}$. $\mathcal{D}_k$ denotes the subset of $\mathcal{D}$ containing all elements $(\mathbf{x}_i, y_i)$ such that $y_i = k$.

Output: The loss J for a randomly generated training episode.

$V \leftarrow \text{RANDOMSAMPLE}(\{1, \ldots, K\}, N_C)$ ▷ Select class indices for episode

for k in $\{1, \ldots, N_C\}$ **do**

$S_k \leftarrow \text{RANDOMSAMPLE}(\mathcal{D}_{V_k}, N_S)$ ▷ Select support examples

$Q_k \leftarrow \text{RANDOMSAMPLE}(\mathcal{D}_{V_k} \setminus S_k, N_Q)$ ▷ Select query examples

$$\mathbf{c}_k \leftarrow \frac{1}{N_C} \sum_{(\mathbf{x}_i, y_i) \in S_k} f_\phi(\mathbf{x}_i)$$ ▷ Compute prototype from support examples

end for

$J \leftarrow 0$ ▷ Initialize loss

for k in $\{1, \ldots, N_C\}$ **do**

for $(\mathbf{x}, y)$ in Q_k **do**

$$J \leftarrow J + \frac{1}{N_C N_Q} \left[d(f_\phi(\mathbf{x}), \mathbf{c}_k)) + \log \sum_{k'} \exp(-d(f_\phi(\mathbf{x}), \mathbf{c}_k)) \right]$$ ▷ Update loss

end for

end for

Few-shot in the boundary

- It is known that in hyperbolic neural networks, embeddings are prone to converge to the boundary of P_k^d- in practice, the effective radius r_{eff}.

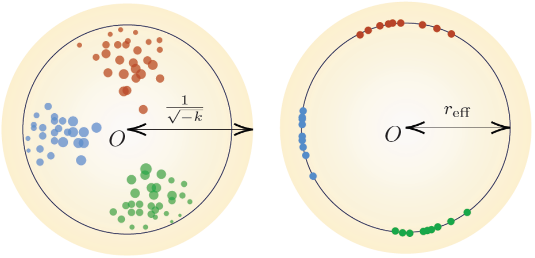


Figure 1. Hyperbolic image embeddings in the Poincaré ball: expectation (left) versus reality (right). In high-dimensional hyperbolic space, the volume of a ball is concentrated near its surface where the hyperbolic metric varies monotonically with the angle. Thus, the hierarchy-revealing property of hyperbolic space is lost.

Hyperbolic prototypical learning

- Assume an image dataset $\mathcal{I}$ with C semantic classes.
- If $\mathcal{M}$ is a manifold and $f_\theta : \mathcal{I} \to \mathcal{M}$ is an image encoder parametrized by θ, a typical approach models the probability of $\mathbf{z}_i \in \mathcal{I}$ being of class c as

$$p(c|\mathbf{z}_i) = \frac{\exp(-d_{\mathcal{M}}(\mathbf{w}_c, f(\mathbf{z}_i;\theta)))}{\sum_k \exp(-d_{\mathcal{M}}(\mathbf{w}_k, f(\mathbf{z}_i;\theta)))} \tag{1}$$

  where $\mathbf{w}_c \in \mathcal{M}$ is the centroid of the c-th class.
- While classic networks use l_2^2 as $d_{\mathcal{M}}$ (related to a Bregman divergence), Poincaré networks use the geodesic distance (4) instead.

- For simplicity, let $\mathbf{x}_i = f(\mathbf{z}_i; \theta)$ and reformulate the objective :

$$L_i := d_{\mathcal{M}}(\mathbf{w}_c, \mathbf{x}_i) + \log \sum_k e^{-d_{\mathcal{D}}(\mathbf{w}_k, \mathbf{x}_i)} \quad (2)$$

$$\mathbf{x} = arg \min_{\mathbf{x}} \sum_i L_i = (arg \min_{\mathbf{x}_1} L_1, \ldots, arg \min_{\mathbf{x}_{|\mathcal{I}|}} L_{|\mathcal{I}|})$$

  That is, we optimize over $\mathbf{x}$ instead of θ.

- Then, the optimal state of $\mathbf{x}$ is obtained by
  - pick C direntions in $\mathbb{R}^d$
  - set the direction of each embedding to match that of its class $\mathbf{x}_i = r\mathbf{w}_c$ for a certain $r > 0$
  - the first term of L_i is automatically zero and the second term goes to $-\infty$ as the embeddings approach the boundary ($r \to 1/\sqrt{-k}$).

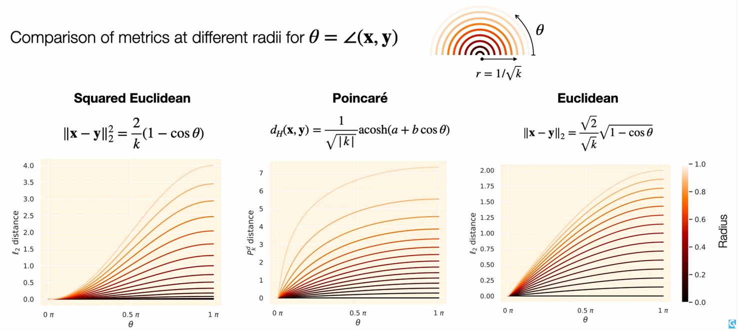
High-dimensional hyperbolic space

Hyperbolic measure concentration

For large d, the volume of a hyperbolic ball is concentrated close to its boundary.

- The proposition leads to the hypothesis : given the high dimensionality of the Poincaré ball used in the hyperbolic few-shot literature, embeddings should lie at, or close to, r_{eff}.
- QUESTION : The hyperbolicity of the learnt representation space????

- In fact, a hyperbolic $(d-1)$-sphere containing embeddings at r_{eff} from the origin is **isometric to an Euclidean** $(d-1)$**-sphere of radius** $\frac{2/\sqrt{-k}+2r_{\text{eff}}}{1/\sqrt{-k}-r_{\text{eff}}}$.
- Plus, Euclidean metric l_2 is not that different from the hyperbolic.



Fixed-radius Euclidean embeddings

- Based on the facts, the authors proposed a fixed-radius Euclidean embeddings with the metric $l_2 \propto \sqrt{1 - \cos(\alpha)}$.
- Given the radius hyperparameter $r = 1/\sqrt{k}$ ($k > 0$ is a shperical curvature) and the Euclidean backbone $f(\cdot; \theta)$, the embeddings fed to the prototypical loss (1) are computed as

$$r \frac{f(\mathbf{x}; \theta)}{\|f(\mathbf{x}; \theta)\|_2}$$

- This Euclidean architecture is ahead of the hyperbolic few-shot SoTA in most of the experiments conducted.

- Used a 4-layer ConvNet as backbone with variable output dimension equal to d. These embeddings were then projected
  - to the Poincaré ball through the exponential map at the origin
  - through magnitude rescaling in the Euclidean case
- In the latter, the radius is a hyperparameter, similarly to the curvature in the former.

Experiments

d	Space	Test acc	$r_{\min}$	r_{avg}	$r_{\max}$
2^7	$(\mathbb{R}^d, \ell_2^2)$	78.95 ± 0.16	-	-	-
	$P^d_{-0.05}$	79.17 ± 0.17	4.35	4.47	4.47
	$P^d_{-0.01}$	82.30 ± 0.15	8.07	9.59	9.87
	(S^d, ℓ_2)	$\mathbf{83.13 \pm 0.14}$	22.36	22.36	22.36
2^8	$(\mathbb{R}^d, \ell_2^2)$	80.14 ± 0.16	-	-	-
	$P^d_{-0.05}$	80.58 ± 0.16	4.33	4.47	4.47
	$P^d_{-0.01}$	84.51 ± 0.14	9.89	9.99	9.99
	(S^d, ℓ_2)	$\mathbf{85.01 \pm 0.14}$	22.36	22.36	22.36
2^9	$(\mathbb{R}^d, \ell_2^2)$	79.95 ± 0.15	-	-	-
	$P^d_{-0.05}$	81.04 ± 0.16	4.46	4.47	4.47
	$P^d_{-0.01}$	84.60 ± 0.14	9.90	9.99	9.99
	(S^d, ℓ_2)	$\mathbf{85.18 \pm 0.14}$	22.36	22.36	22.36
2^{10}	$(\mathbb{R}^d, \ell_2^2)$	78.83 ± 0.15	-	-	-
	$P^d_{-0.05}$	81.06 ± 0.16	4.47	4.47	4.47
	$P^d_{-0.01}$	84.70 ± 0.14	9.99	9.99	9.99
	(S^d, ℓ_2)	$\mathbf{85.37 \pm 0.14}$	22.36	22.36	22.36

Table 2. **CUB_200_2011 5-shot 5-way** test results, 95% confidence intervals and embedding radii.

d	Space	Test acc	$r_{\min}$	r_{avg}	$r_{\max}$
2^7	$(\mathbb{R}^d, \ell_2^2)$	49.11 ± 0.20	-	-	-
	$P^d_{-0.005}$	49.10 ± 0.20	3.78	5.85	7.41
	$P^d_{-0.01}$	49.60 ± 0.20	2.78	3.53	4.15
	(S^d, ℓ_2)	$\mathbf{50.24 \pm 0.20}$	22.36	22.36	22.36
2^8	$(\mathbb{R}^d, \ell_2^2)$	49.14 ± 0.20	-	-	-
	$P^d_{-0.005}$	49.25 ± 0.20	8.77	9.93	10.44
	$P^d_{-0.01}$	47.07 ± 0.19	7.54	8.05	9.43
	(S^d, ℓ_2)	$\mathbf{50.36 \pm 0.20}$	22.36	22.36	22.36
2^9	$(\mathbb{R}^d, \ell_2^2)$	48.84 ± 0.20	-	-	-
	$P^d_{-0.005}$	45.59 ± 0.18	14.12	14.13	14.13
	$P^d_{-0.01}$	48.71 ± 0.19	9.98	9.99	9.99
	(S^d, ℓ_2)	$\mathbf{50.97 \pm 0.19}$	22.36	22.36	22.36
2^{10}	$(\mathbb{R}^d, \ell_2^2)$	49.10 ± 0.20	-	-	-
	$P^d_{-0.005}$	49.19 ± 0.19	14.13	14.13	14.13
	$P^d_{-0.01}$	$\mathbf{51.37 \pm 0.20}$	9.99	9.99	9.99
	(S^d, ℓ_2)	51.36 ± 0.20	22.36	22.36	22.36

Table 3. **MiniImageNet 1-shot 5-way** test results, 95% confidence intervals and embedding radii.

Appendix

Hyperbolic Space

- Hyperboloid model

$$H_k^d := \{\mathbf{x} \in \mathbb{R}^{d,1} | \langle \mathbf{x}, \mathbf{x} \rangle_L = \frac{1}{k}, x_{d+1} > 0\}, \text{ with curvature } k < 0$$

$$\langle \mathbf{x}, \mathbf{x} \rangle_L := \sum_{i=1}^{d} x_i y_i - x_{d+1} y_{d+1}, \quad \mathbb{R}^{d,1} := \{\mathbf{x} = (x_1, \ldots, x_{d+1}) \in \mathbb{R}^d \times \mathbb{R}\}$$

- Poincaré Model

$$B_k^d = \{x \in \mathbb{R}^d : \|\mathbf{x}\|_2^2 < -\frac{1}{k}\}, \ k < 0$$

- Poincaré ball can be derived from the hyperboloid model as follows.

$$\Pi : H_k^d \to P_k^d$$

$$\Pi(\mathbf{x}) := \left(\frac{x_1}{1 + \sqrt{-k} x_{d+1}}, \ldots, \frac{x_d}{1 + \sqrt{-k} x_{d+1}} \right) \tag{3}$$

Hyperbolic Space

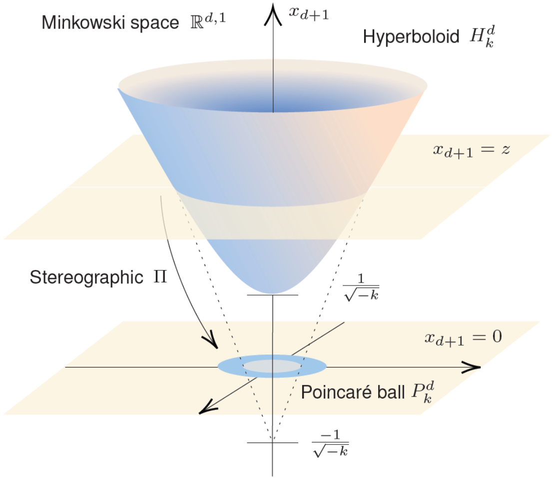


Figure 2. Minkowski ambient space $\mathbb{R}^{d,1}$, hyperboloid H_k^d, stereographic projection Π and Poincaré ball model P_k^d.

- The inverse projection of (3), $\Pi^{-1} : P_k^d \to H_k^d$ takes the form

$$\Pi^{-1}(\mathbf{u}) = \left(\lambda(\mathbf{u})\mathbf{u}, \frac{1}{\sqrt{-k}}(\lambda(\mathbf{u}) - 1)\right)$$

$$\lambda(\mathbf{u}) = 2/\left(1 + k\|\mathbf{u}\|_2^2\right)$$

- The Poincaré exponential map at the origin

$$\mathrm{Exp}_{\mathbf{0}}(\mathbf{v}) = \tanh\left(\sqrt{-k}\|\mathbf{v}\|_2\right) \frac{\mathbf{v}}{\sqrt{-k}\|\mathbf{v}\|_2}$$

projects a tangent vector back to the ball.

Note that since Poincaré ball is open, the tangent space at each point is simply isomorphic to $\mathbb{R}^d$. On the other hand, hyperboloid has an explicit form of a tangent space at each point.

- Poincaré geodesic distance between any **x** and $\mathbf{y} \in P_k^d$ is given by

$$d_{P_k^d}(\mathbf{x}, \mathbf{y}) := \frac{2}{\sqrt{-k}} \operatorname{arctanh}(\sqrt{-k} \| -\mathbf{x} \oplus \mathbf{y} \|_2) \tag{4}$$

$$\text{where } \mathbf{x} \oplus \mathbf{y} = \frac{(1 - 2k\langle \mathbf{x}, \mathbf{y} \rangle - k\|\mathbf{y}\|_2^2)\mathbf{x} + (1 + k\|\mathbf{x}\|_2^2)\mathbf{y}}{1 - 2k\langle \mathbf{x}, \mathbf{y} \rangle + k^2\|\mathbf{x}\|_2^2\|\mathbf{y}\|_2^2}$$

- Poincaré image encoder

  Given an Euclidean backbone f with parameters θ, the hyperbolic image encoders embeds an image **x** as follows:

$$h(\mathbf{x}; \theta) = \operatorname{Exp}_0^P(f(\mathbf{x}; \theta))$$