Supplementary for Clipped Hyperbolic Classifiers Are Super-Hyperbolic Classifiers

Yunhui Guo[1] Xudong Wang[1] Yubei Chen[2] Stella X. Yu[1]
[1]UC Berkeley / ICSI [2]Facebook AI Research

1. Organization of the supplementary

1.1. Background and datasets

1. In Section 2, we introduce the background on Gyrovector space.
2. In Section 4, we give the detailed statistics of the datasets.

1.2. Effect of the gradient update

In Section 3, we derive the effect of the gradient update of the Euclidean parameters on the hyperbolic embeddings.

1.3. Effect of the clipped value r

In Section 5, we show the effect of different choices of r. *To conclude, there is a sweet spot in terms of choosing r which is neither too large (causing vanishing gradient problem) nor too small (not enough capacity). The performance of clipped HNNs is also robust to the choice of the hyperparameter r if it is around the sweet spot.*

1.4. Adversarial robustness

In Section 6, we show more results on adversarial robustness. *Clipped HNNs show more robustness to ENNs and greatly improve the robustness of vanilla HNNs.*

1.5. More on out-of-distribution detection

In Section 7, we show more results on out-of-distribution (OOD) detection using ENNs, vanilla HNNs and clipped HNNs. *Clipped HNNs show stronger OOD detection capability compared with ENNs and greatly improve the OOD detection capability of vanilla HNNs.*

1.6. Using softmax with temperature scaling as a workaround

In Section 8 we show that using softmax with temperature scaling as a workaround for addressing the vanishing gradient problem. *When feature dimension is high, softmax with temperature scaling severely underperforms the proposed feature clipping. The results again confirm the effectiveness of the proposed approach.*

1.7. Clipped hyperbolic space is still hyperbolic

In Section 9, we show that clipped hyperbolic space is still hyperbolic. *Using clipped hyperbolic space for learning word embeddings is better than using the unclipped version.*

1.8. With norm regularization term

In Section 10, we show the results using norm regularization during training. *The proposed feature clipping outperforms using the norm regularization term.*

1.9. More discussions on Lorentz model

In Section 11, we give more discussions on Lorentz model and why we focus on hyperbolic neural networks based on Poincaré ball model.

2. Gyrovector space

We give more details on gyrovector space, for a more systematic treatment, please refer to [6–8].

Gyrovector space provides a way to operate in hyperbolic space with vector algebra. Gyrovector space to hyperbolic geometry is similar to standard vector space to Euclidean geometry. The geometric objects in gyrovector space are called gyroevectors which are equivalent classes of directed gyrosegments. Similar to the vectors in Euclidean space which are added according to parallelogram law, gyrovectors are added according to gyroarallelogram law. Technically, gyrovector spaces are gyrocommutative gyrogroups of gyrovectors that admit scalar multiplications.

We start from the introduction of gyrogroups which give rise to gyrovector spaces.

Definition 2.1 (Gyrogroups) *A groupoid $(G, \oplus)$ is a gyrogroup if it satisfies the follow axioms,*

1. *There exist one element $0 \in G$ satisfies $0 \oplus a = a$ for all $a \in G$.*
2. *For each $a \in G$, there exist an element $\ominus a \in G$ which satisfies $\ominus a \oplus a = 0$*
3. *For every $a, b, c \in G$, there exist a unique element $gyr[a, b]c \in G$ such that $\oplus$ satisfies the left gyroassociative law $a \oplus (b \oplus c) = (a \oplus b) \oplus gyr[a, b]c$.*
4. *The map $gyr[a, b]c$: $G \to G$ given by $c \mapsto gyr[a, b]c$ is an automorphism of the groupoid $(G, \oplus)$: $gyr[a, b] \in Aut(G, \oplus)$. The automorphism $gyr[a, b]$ of G is called the gyroautomorphism of G generated by $a, b \in G$.*
5. *The operation gry: $G \times G \to Aut(G, \oplus)$ is called gyrator of G. The gyroautomorphism $gyr[a, b]$ generated by any $a, b \in G$ has the left loop property: $gyr[a, b] = gyr[a \oplus b, b]$.*

In particular, Möbius complex disk groupoid $(\mathbb{D}, \oplus_M)$ is a gyrocommunicative gyrogroup, where $\mathbb{D} = \{z \in \mathbb{C} : |z| < 1\}$ and $\oplus_M$ is the Möbius addition. The same applies to the s-ball $\mathbb{V}_s$ which is defined as,

$$\mathbb{V}_s = \{\mathbf{v} \in \mathbb{V} : \|\mathbf{v}\| < s\} \tag{1}$$

Gyrocommutative gyrogroups which admit scalar multiplication $\oplus$ become gyrovector space $(G, \oplus, \otimes)$. Möbius gyrogroups $(\mathbb{V}, \oplus_M)$ admit scalar multiplication $\oplus_M$ become Möbius gyrovector space $(\mathbb{V}, \oplus_M, \otimes_M)$.

Definition 2.2 (Möbius Scalar Multiplication) *Let $(\mathbb{V}_s, \oplus_M)$ be a Möbius gyrogroup, the Möbius scalar multiplication $\otimes_M$ is defined as,*

$$r \otimes_M \mathbf{v} = s \frac{(1 + \frac{\|\mathbf{v}\|}{s})^r - (1 - \frac{\|\mathbf{v}\|}{s})^r}{(1 + \frac{\|\mathbf{v}\|}{s})^r + (1 - \frac{\|\mathbf{v}\|}{s})^r} \frac{\mathbf{v}}{\|\mathbf{v}\|} \tag{2}$$

where $r \in \mathbb{R}$ and $\mathbf{v} \in \mathbb{V}_s$, $\mathbf{v} \neq \mathbf{0}$.

Definition 2.3 (Gyrolines) *Let $\mathbf{a}, \mathbf{b}$ be two distinct points in the gyrovector space $(G, \oplus, \otimes)$. The gyroline in G which passes through $\mathbf{a}, \mathbf{b}$ is the set of points:*

$$L = \mathbf{a} \oplus (\ominus \mathbf{a} \oplus \mathbf{b}) \otimes t \tag{3}$$

where $t \in \mathbb{R}$.

It can be proven that gyrolines in a Möbius gyrovector space coincide with the geodesics of the Poincaré ball model of hyperbolic geometry.

With the aid of operations in gyrovector spaces, we can define important properties of the Poincaré ball model in closed-form expressions.

Definition 2.4 (Exponential Map and Logarithmic Map) *As shown in [1], the exponential map $\exp^c_{\mathbf{x}} : T_{\mathbf{x}}\mathbb{B}^n_c \to \mathbb{B}^n_c$ is defined as,*

$$\exp^c_{\mathbf{x}}(\mathbf{v}) = \mathbf{x} \oplus_c (\tanh(\frac{\sqrt{c}\lambda^c_{\mathbf{x}}\|\mathbf{v}\|}{2})\frac{\mathbf{v}}{\sqrt{c}\|\mathbf{v}\|}), \qquad \forall \mathbf{x} \in \mathbb{B}^n_c, \mathbf{v} \in T_{\mathbf{x}}\mathbb{B}^n_c \tag{4}$$

The logarithmic map $\log^c_{\mathbf{x}} : \mathbb{B}^n_c \to T_{\mathbf{x}}\mathbb{B}^n_c$ is defined as,

$$\log^c_{\mathbf{x}} = \frac{2}{\sqrt{c}\lambda^c_{\mathbf{x}}} \tanh^{-1}(\sqrt{c}\|\ominus_c \mathbf{x} \oplus_c \mathbf{y}\|)\frac{\ominus_c \mathbf{x} \oplus_c \mathbf{y}}{\|\ominus_c \mathbf{x} \oplus_c \mathbf{y}\|}, \qquad \mathbf{x}, \mathbf{y} \in \mathbb{B}^n_c \tag{5}$$

The distance between two points in the Poincaré ball can be defined as,

Definition 2.5 *(Poincaré Distance between Two Points)*

$$d_c(\mathbf{x}, \mathbf{y}) = \frac{2}{\sqrt{c}} \tanh^{-1}(\sqrt{c}\|\ominus_c \mathbf{x} \oplus_c \mathbf{y}\|) \tag{6}$$

3. The Effect of gradient update of Euclidean parameters on the hyperbolic embedding

We derive the effect of the a single gradient update of the Euclidean parameters on the hyperbolic embedding. For the Euclidean sub-network $E : \mathbb{R}^m \to \mathbb{R}^n$. Consider the first-order Taylor-expansion of the Euclidean network with a single gradient update,

$$\begin{aligned} E(\mathbf{w}^E_{t+1}) &= E(\mathbf{w}^E_t + \eta \frac{\partial \ell}{\partial \mathbf{w}^E}) \\ &\approx E(\mathbf{w}^E_t) + \eta (\frac{\partial E(\mathbf{w}^E_t)}{\partial \mathbf{w}^E_t})^T \frac{\partial \ell}{\partial \mathbf{w}^E} \end{aligned} \tag{7}$$

Meanwhile, the exponentional map of the Poincaré ball is,

$$\mathrm{Exp}^c_{\mathbf{0}}(\mathbf{v}) = \tanh(\sqrt{c}\|\mathbf{v}\|)\frac{\mathbf{v}}{\sqrt{c}\|\mathbf{v}\|} \tag{8}$$

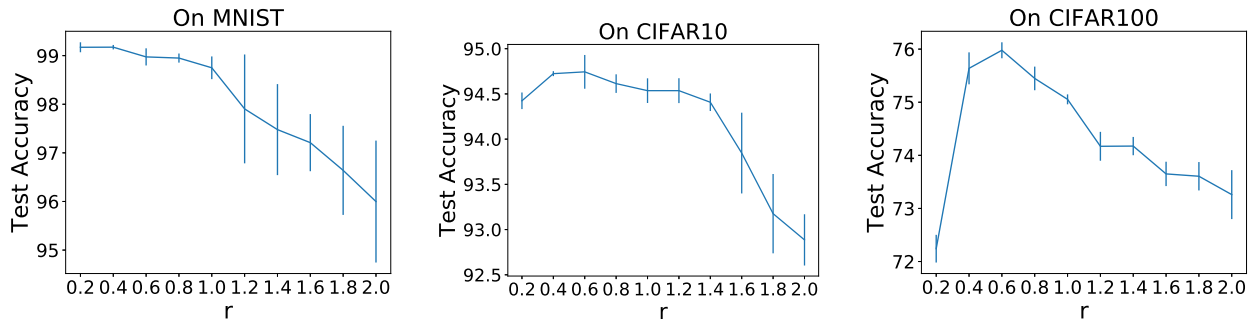

Figure 1. We show the change of the test accuracy as we vary the hyperparameter r. A large r leads to *vanishing gradient problem* and a small r causes insufficient capacity. Both lead to a drop in test accuracy.

Table 1. **The results of out-of-distribution detection** on CIFAR10 with softmax score. The results of ENNs are shaded in dark gray. The results of vanilla HNNs are shaded in light gray. Clipped HNNs achieve higher average performance across all the datasets and greatly improve the OOD detection capability of vanilla HNNs.

OOD Dataset	FPR95 ↓	AUROC ↑	AUPR ↑
ISUN	46.30 ± 0.78	91.50 ± 0.16	98.16 ± 0.05
	98.37 ± 0.20	29.72 ± 0.58	74.54 ± 0.19
	45.28 ± 0.65	91.61 ± 0.21	98.09 ± 0.06
Place365	51.09 ± 0.92	87.56 ± 0.37	96.76 ± 0.15
	96.10 ± 0.32	44.82 ± 0.65	80.67 ± 0.34
	54.77 ± 0.76	86.82 ± 0.41	96.17 ± 0.20
Texture	65.04 ± 0.91	82.80 ± 0.35	94.59 ± 0.20
	97.62 ± 0.15	33.87 ± 0.40	74.52 ± 0.17
	47.12 ± 0.62	89.91 ± 0.20	97.39 ± 0.09
SVHN	71.66 ± 0.84	86.58 ± 0.21	97.06 ± 0.06
	91.03 ± 0.53	61.33 ± 0.53	86.39 ± 0.27
	49.89 ± 1.03	91.34 ± 0.22	98.13 ± 0.06
LSUN-Crop	22.22 ± 0.78	96.05 ± 0.10	99.16 ± 0.03
	96.18 ± 0.36	37.29 ± 0.63	75.94 ± 0.27
	23.87 ± 0.73	95.65 ± 0.22	98.98 ± 0.07
LSUN-Resize	41.06 ± 1.07	92.67 ± 0.16	98.42 ± 0.04
	99.62 ± 0.10	22.05 ± 0.32	71.88 ± 0.15
	41.49 ± 1.24	92.97 ± 0.24	98.46 ± 0.07
Mean	49.56	89.53	97.36
	96.49	38.18	77.32
	43.74	**91.38**	**97.87**

Table 2. **The results of out-of-distribution detection** on CIFAR100 with softmax score. The results of ENNs are shaded in dark gray. The results of vanilla HNNs are shaded in light gray. Clipped HNNs achieve comparable performance to ENNs in terms of AUPR and higher average performance in terms of PRR95 and AUROC.

OOD Dataset	FPR95 ↓	AUROC ↑	AUPR ↑
ISUN	74.07 ± 0.87	82.51 ± 0.39	95.83 ± 0.11
	80.97 ± 0.65	69.24 ± 0.52	89.98 ± 0.22
	68.37 ± 0.90	81.31 ± 0.43	94.96 ± 0.20
Place365	81.01 ± 1.07	76.90 ± 0.45	94.02 ± 0.15
	82.75 ± 0.66	71.97 ± 0.50	92.27 ± 0.22
	79.66 ± 0.69	76.94 ± 0.28	93.91 ± 0.18
Texture	83.67 ± 0.68	77.52 ± 0.32	94.47 ± 0.10
	75.33 ± 0.92	75.14 ± 0.49	92.39 ± 0.20
	64.91 ± 0.80	83.26 ± 0.25	95.77 ± 0.08
SVHN	84.56 ± 0.78	84.32 ± 0.22	96.69 ± 0.07
	62.83 ± 0.70	84.29 ± 0.23	96.31 ± 0.06
	53.11 ± 1.04	89.53 ± 0.26	97.71 ± 0.07
LSUN-Crop	43.46 ± 0.79	93.09 ± 0.23	98.58 ± 0.05
	56.66 ± 0.67	89.30 ± 0.17	97.71 ± 0.04
	51.08 ± 1.17	87.21 ± 0.39	96.83 ± 0.13
LSUN-Resize	71.50 ± 0.73	82.12 ± 0.40	95.69 ± 0.13
	75.50 ± 0.81	73.40 ± 0.68	91.41 ± 0.28
	63.86 ± 1.10	82.36 ± 0.42	95.16 ± 0.13
Mean	73.05	82.74	**95.88**
	72.34	77.22	93.35
	63.50	**83.43**	95.72

The gradient of the exponential map can be computed as,

$$\begin{aligned}\nabla \mathrm{Exp}_{\mathbf{0}}^{c}(\mathbf{v}) &= \frac{\mathbf{v}}{\sqrt{c}\|\mathbf{v}\|}\nabla \tanh(\sqrt{c}\|\mathbf{v}\|) + \tanh(\sqrt{c}\|\mathbf{v}\|)\nabla\frac{\mathbf{v}}{\sqrt{c}\|\mathbf{v}\|} \\ &= 1 - \tanh^2(\sqrt{c}\|\mathbf{v}\|) + \tanh(\sqrt{c}\|\mathbf{v}\|)\frac{1}{\sqrt{c}}\frac{2}{\|\mathbf{v}\|}\end{aligned} \tag{9}$$

Let $\mathbf{x}_{t+1}^H$ be the projected point in hyperbolic space, i.e,

$$\mathbf{x}_{t+1}^H = \mathrm{Exp}_{\mathbf{0}}^{c}(E(\mathbf{w}_{t+1}^E)) \tag{10}$$

Again we can apply the first-order Taylor-expansion on the exponential map,

$$\begin{aligned}\mathbf{x}_{t+1}^H &= \mathrm{Exp}_{\mathbf{0}}^{c}(E(\mathbf{w}_{t+1}^E)) \\ &\approx \mathrm{Exp}_{\mathbf{0}}^{c}(E(\mathbf{w}_{t}^E) + \eta(\frac{\partial E(\mathbf{w}_t^E)}{\partial \mathbf{w}_t^E})^T\frac{\partial \ell}{\partial \mathbf{w}^E})\end{aligned} \tag{11}$$

Table 3. **The results of out-of-distribution detection** on CIFAR10 with energy score. The results of ENNs are shaded in dark gray. The results of vanilla HNNs are shaded in light gray.

OOD Dataset	FPR95 ↓	AUROC ↑	AUPR ↑
ISUN	34.19 ± 0.97	93.07 ± 0.24	98.42 ± 0.07
	99.31 ± 0.15	28.69 ± 0.35	74.14 ± 0.10
	25.39 ± 0.32	95.48 ± 0.09	99.01 ± 0.04
Place365	43.34 ± 1.22	88.50 ± 0.48	96.76 ± 0.17
	97.57 ± 0.37	43.96 ± 0.87	80.36 ± 0.46
	45.17 ± 1.19	89.61 ± 0.28	97.20 ± 0.14
Texture	58.51 ± 0.77	82.98 ± 0.20	94.55 ± 0.14
	95.93 ± 0.29	35.02 ± 0.41	74.87 ± 0.14
	49.70 ± 0.94	90.66 ± 0.20	97.98 ± 0.04
SVHN	49.04 ± 1.05	91.57 ± 0.13	98.12 ± 0.05
	96.71 ± 0.37	59.65 ± 0.56	86.16 ± 0.25
	57.33 ± 1.34	88.45 ± 0.20	97.44 ± 0.06
LSUN-Crop	9.48 ± 0.60	98.21 ± 0.07	99.63 ± 0.02
	98.18 ± 0.27	36.34 ± 0.63	75.64 ± 0.26
	24.78 ± 0.73	95.06 ± 0.15	98.92 ± 0.05
LSUN-Resize	28.28 ± 0.66	94.31 ± 0.14	98.72 ± 0.04
	99.91 ± 0.06	21.34 ± 0.48	71.60 ± 0.21
	22.52 ± 0.67	96.15 ± 0.09	99.18 ± 0.02
Mean	**37.14**	91.44	97.70
	97.94	37.50	77.13
	37.48	**92.57**	**98.29**

Table 4. **The results of out-of-distribution detection** on CIFAR100 with energy score. The results of ENNs are shaded in dark gray. The results of vanilla HNNs are shaded in light gray.

OOD Dataset	FPR95 ↓	AUROC ↑	AUPR ↑
ISUN	74.49 ± 0.60	82.45 ± 0.33	95.84 ± 0.12
	81.73 ± 0.54	70.38 ± 0.28	90.76 ± 0.20
	68.75 ± 0.93	81.33 ± 0.31	94.93 ± 0.16
Place365	81.20 ± 0.86	77.02 ± 0.34	94.13 ± 0.13
	82.73 ± 0.98	74.04 ± 0.55	93.20 ± 0.24
	79.51 ± 0.69	77.23 ± 0.37	93.97 ± 0.17
Texture	83.19 ± 0.31	77.74 ± 0.35	94.54 ± 0.11
	72.77 ± 0.52	77.38 ± 0.39	93.38 ± 0.21
	65.03 ± 0.52	83.38 ± 0.29	95.85 ± 0.10
SVHN	84.12 ± 0.59	84.41 ± 0.16	96.72 ± 0.04
	53.37 ± 0.67	86.37 ± 0.30	96.78 ± 0.08
	55.44 ± 1.00	89.43 ± 0.25	97.69 ± 0.06
LSUN-Crop	43.80 ± 1.29	93.04 ± 0.22	98.56 ± 0.05
	87.32 ± 0.36	83.09 ± 0.20	96.40 ± 0.05
	74.89 ± 0.73	84.98 ± 0.18	96.46 ± 0.08
LSUN-Resize	71.86 ± 0.69	81.86 ± 0.27	95.60 ± 0.09
	81.81 ± 0.71	72.96 ± 0.59	91.64 ± 0.23
	64.35 ± 0.62	82.64 ± 0.36	95.27 ± 0.14
Mean	73.11	82.75	**95.90**
	76.62	77.37	93.69
	67.99	**83.17**	95.70

Denote $\eta(\frac{\partial E(\mathbf{w}_t^E)}{\partial \mathbf{w}_t^E})^T \frac{\partial \ell}{\partial \mathbf{w}^E}$ by $J_{\mathbf{w}_t^E}$, we have

$$\begin{aligned}\mathbf{x}_{t+1}^H &= \mathrm{Exp}_{\mathbf{0}}^c(E(\mathbf{w}_{t+1}^E)) \\ &\approx \mathrm{Exp}_{\mathbf{0}}^c(E(\mathbf{w}_t^E) + J_{\mathbf{w}_t^E}) \\ &\approx \mathrm{Exp}_{\mathbf{0}}^c(E(\mathbf{w}_t^E)) + (\frac{\partial \mathrm{Exp}_{\mathbf{0}}^c(E(\mathbf{w}_t^E))}{\partial E(\mathbf{w}_t^E)})^T J_{\mathbf{w}_t^E} \\ &= \mathbf{x}_t^H + (\frac{\partial \mathrm{Exp}_{\mathbf{0}}^c(E(\mathbf{w}_t^E))}{\partial E(\mathbf{w}_t^E)})^T J_{\mathbf{w}_t^E}\end{aligned} \tag{12}$$

Denote $(\frac{\partial \mathrm{Exp}_{\mathbf{0}}^c(E(\mathbf{w}_t^E))}{\partial E(\mathbf{w}_t^E)})^T \eta(\frac{\partial E(\mathbf{w}_t^E)}{\partial \mathbf{w}_t^E})^T$ by $C(E(\mathbf{w}_t^E))$,

$$\mathbf{x}_{t+1}^H = \mathbf{x}_t^H + C(E(\mathbf{w}_t^E))^T \frac{\partial \ell}{\partial \mathbf{w}^E} \tag{13}$$

4. Datasets

The statistics of the datasets are shown in Table 5.

	MNIST	CIFAR10	CIFAR100	ImageNet
# of Training Examples	60,000	50,000	50,000	1,281,167
# of Test Examples	10,000	10,000	10,000	50,000

Table 5. The statistics of the datasets.

5. The effect of hyperparameter r

We conduct ablation studies to show the effect of the hyperparameter r which is the maximum norm of the Euclidean embedding. In Figure 1 we show the change of test accuracy as we vary the hyperparameter r on MNIST, CIFAR10 and CIFAR100. We repeat the experiments for each choice of r five times and report both average accuracy and standard deviation. On the one hand, it can be observed that a larger r leads to a drop in test accuracy. As we point out, this is caused by the vanishing gradient problem in training hyperbolic neural networks. On the other hand, a small r can also lead to a drop in test accuracy especially on more complex tasks such as CIFAR10 and CIFAR100. The plausible reason is that a small r reduces the capacity of the embedding space which is detrimental for learning discriminative features.

To conclude, there is a sweet spot in terms of choosing r which is neither too large (causing vanishing gradient problem) nor too small (not enough capacity). The performance of clipped HNNs is also robust to the choice of the hyperparameter r if it is around the sweet spot.

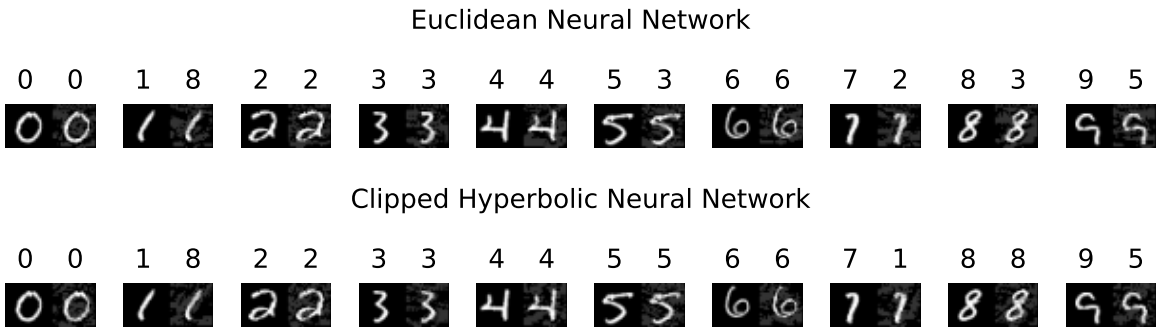

Figure 2. Clipped HNNs show more adversarial robustness compared with ENNs. We show the clean image and the corresponding adversarial image and the predictions of the network of 10 randomly sampled images. In several cases, clipped HNNs make correct predictions on the adversarial images while Euclidean neural networks make wrong predictions.

6. More results on adversarial robustness

Although we observe that with adversarial training, hyperbolic neural networks achieve similar robust accuracy to Euclidean neural networks, in a further study, we consider training models using a small ϵ but attacking with a larger ϵ with FGSM on MNIST. In Table 6 we show the results of training the networks using $\epsilon = 0.05$ and attacking with ϵ = 0.1, 0.2 and 0.3. We can observe that for attacking with larger ϵ such as 0.2 and 0.3, clipped HNNs show more robustness to ENNs. Moreover, clipped HNNs greatly improve the robustness of vanilla HNNs. The possible explanation is that the proposed feature clipping reduces the adversarial noises in the forward pass and also improve the performance of vanilla HNNs. One of the future directions is to systematically understand and analyze the reason behind the robustness of clipped HNNs. In Figure 2, we show the clean and adversarial images generated by FGSM with clipped HNNs and ENNs respectively. The predictions of the networks are shown above the image. It can be observed that clipped HNNs show more adversarial robustness compared with ENNs.

Network \ ϵ	0.1	0.2	0.3
ENNs	94.51 $\pm$ 0.32 %	67.85 $\pm$ 2.12 %	42.18 $\pm$ 1.32 %
Vanilla HNNs	81.08 $\pm$ 4.03 %	46.57 $\pm$ 2.09 %	17.21 $\pm$ 3.27%
Clipped HNNs	93.34 $\pm$ 0.16 %	74.97 $\pm$ 1.02 %	46.27 $\pm$ 1.88 %

Table 6. Adversarial training with FGSM (ϵ = 0.05) on MNIST. For attacking with larger ϵ such as 0.1, 0.2 and 0.3, clipped HNNs greatly improve the robustness of vanilla HNNs and show more robustness to ENNs when the attacking with large perturbations (ϵ = 0.2 and ϵ = 0.3).

7. More results on out-of-distribution detection (OOD)

7.1. Results with energy score

In Table 3 and 4 we show the results of using energy score [3] on CIFAR10 and CIFAR100 for out-of-distribution detection. We can observe that on CIFAR10, clipped HNNs achieve comparable performance in terms of FPR95 and perform much better in terms other AUROC and AUPR compared with ENNs. On CIFAR100, clipped HNNs achieve comparable performance in terms of AUPT and perform much better in terms other FPR95 and AUROC compared with ENNs. The results are consistent with the case of using softmax score.

7.2. Clipped HNNs greatly improve the OOD detection capability of vanilla HNNs.

In Table 1 - Table 4, we also show the results of using vanilla HNNs for out-of-distribution detection. Across all the datasets and scores, we can see that clipped HNNs greatly improve the OOD detection capability of vanilla HNNs. This shows that vanilla HNNs have poor OOD detection ability which can greatly limit their practical applicability.

8. Softmax with temperature scaling

We consider softmax with temperature scaling as an alternative for addressing the vanishing gradient problem in training hyperbolic neural networks. Softmax with temperature scaling introduces an additional temperature parameter T to adjust the logits before applying the softmax function. Softmax with temperature scaling can be formulated as,

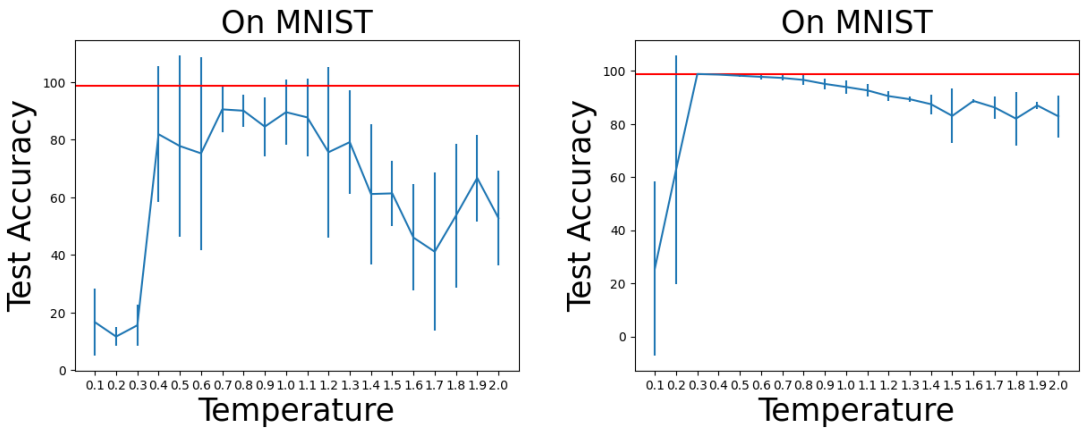

Figure 3. We show the change of the test accuracy as we vary the temperature parameter T. The red horizontal line is the result of the hyperbolic neural networks with the proposed feature clipping. Softmax with temperature scaling with a carefully tuned temperature can approach the performance of the proposed feature clipping. However, it is sensitive to the feature dimension and the temperature parameter. **Left**: the embedding dimension is 2. **Right**: the embedding dimension is 64.

$$\text{Softmax}(\boldsymbol{Z}/T)_i = \frac{e^{Z_i/T}}{\sum_{j=1}^{K} e^{Z_j/T}} \quad \text{for} \quad i = 1, ..., K, \quad \boldsymbol{Z} = (Z_1, ..., Z_K) \tag{14}$$

In hyperbolic neural networks, $\boldsymbol{Z}$ is the output of the hyperbolic fully-connected layer and K is the number of classes. If the additional temperature parameter T is smaller than 1, the magnitude (in the Euclidean sense) of the hyperbolic embedding will be scaled up which prevents it from approaching the boundary of the ball.

In Figure 3, we show the performance of training hyperbolic neural networks with temperature scaling compared with the proposed feature clipping. We consider feature dimensions of 2 and 64 respectively. Different temperature parameters are considered and the experiments are repeated for 10 times with different random seeds. We show both the average accuracy and the standard deviation. We can observe that softmax with temperature scaling and a carefully tuned temperature parameter can approach the performance of the proposed feature clipping when the feature dimension is 2. However, the feature dimension is 64, softmax with temperature scaling severely underperforms the proposed feature clipping. The results again confirm the effectiveness of the proposed approach.

9. A magnitude-clipped hyperbolic space is still hyperbolic

The metric in the hyperbolic space with the clipping strategy is still drastically different from that in the Euclidean space, even with magnitude clipping. For the first example, consider two points: a = [0.5, 0.55], b = [0.3, -0.6], the magnitude of both points is smaller than 0.76. The hyperbolic distance between the two points is 3.1822 while the Euclidean distance is 1.1673. This is a two-dimensional example, with a larger embedding dimension, the difference will be much more significant. In Figure 4: left, we compare the hyperbolic line segment with Euclidean line segment given the point a and the point b.

A magnitude-clipped hyperbolic space is still hyperbolic, as the hyperbolic geometry still holds: unlike Euclidean triangles, where the angles always add up to π radians (180°, a straight angle), in hyperbolic geometry the sum of the angles of a hyperbolic triangle is always strictly less than π radians (180°, a straight angle). The difference is referred to as the defect. For a second example, consider three points: A = [0.5, 0.55], B = [0.3, -0.6], C = [-0.1, 0.1]. Their magnitude are all smaller than 0.76. For the triangle ABC, the defect is 58.21° in hyperbolic space and 0° in Euclidean space. This again shows that the clipped hyperbolic space still well maintains the hyperbolic property. In Figure 4: right, we compare the hyperbolic triangle with Euclidean triangle given the point A, the point B and the point C.

We apply the proposed clipping strategy to learn word embedding as in [4]. We perform the reconstruction task on the transitive closure of the WordNet noun hierarchy. We compare the embedding quality of the Euclidean space, the hyperbolic space, clipped hyperbolic space using mean average precision (mAP). The embedding dimension is 10. The results are summarized in Table 7.

We have two conclusions here. First, learning word embeddings with hyperbolic space provides better results than learning in Euclidean space. Second, using hyperbolic space with clipping is slightly better than using hyperbolic space without clipping.

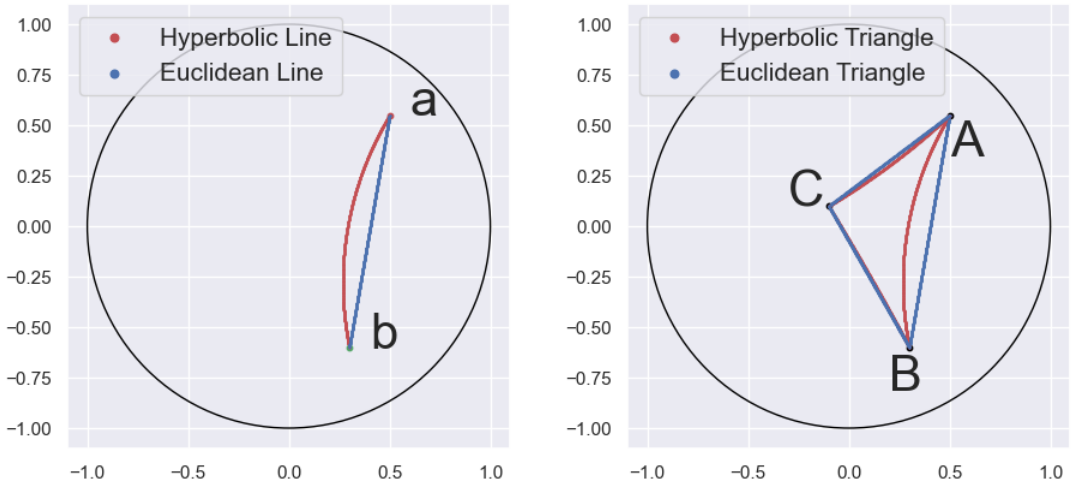

Figure 4. A magnitude-clipped hyperbolic space is still hyperbolic and behaves drastically different from Euclidean space. **Left:** the comparison of hyperbolic line segment with Euclidean line segment. **Right:** the comparison of hyperbolic triangle with Euclidean triangle.

Method	mAP
Euclidean space	0.059
Clipped hyperbolic space	0.860
Hyperbolic space	0.851

Table 7. Learning with word embeddings with clipped hyperbolic space outperforms both with Euclidean space and vanilla hyperbolic space.

10. Additional regularization to minimize the norm of the Euclidean embedding during training

The results of using the regularization term are shown in Table 8.

Method	On CIFR10	On CIFAR100
vanilla HNN	88.82	72.26
w/ regularization	92.71	73.34
w/ clipping	94.76	75.88

Table 8. The proposed feature clipping outperforms vanilla HNNs and HNNs with regularization.

We can see that the clipping strategy outperforms the regularization approach. The reason is that with regularization, the loss function consists of two terms: one is the cross-entropy loss and the other is the regularization loss. It is difficult to balance the two terms . During training, if the cross-entropy loss becomes small, the optimization focuses on minimizing the embedding norm, however a small embedding norm is also detrimental to the performance.

11. More discussions on Lorentz model

Lorentz model is another commonly used model for hyperbolic space. The Lorentz model of n-dimensional hyperbolic space is defined as,

$$\mathcal{H}^n = \{\mathbf{x} \in \mathbb{R}^{n+1} : \langle \mathbf{x}, \mathbf{x} \rangle_{\mathcal{L}} = -1, x_0 > 0\} \tag{15}$$

where $\langle \mathbf{x}, \mathbf{x} \rangle_{\mathcal{L}}$ is the *Lorentzian scalar product* which is defined as,

$$\langle \mathbf{x}, \mathbf{y} \rangle_{\mathcal{L}} = -x_0 y_0 + \sum_{i=1}^{n} x_i y_i \tag{16}$$

The distance function of Lorentz model is given as,

$$d_{\mathcal{H}}(\mathbf{x}, \mathbf{y}) = \operatorname{arccosh}(-\langle \mathbf{x}, \mathbf{y} \rangle_{\mathcal{L}}) \tag{17}$$

Lorentz model is used recently to overcome the numerical issues of the distance function in Poincaré ball model for learning word embeddings [5]. However, it is most effective only in low dimensions [2]. For image datasets of ImageNet-scale, hyperbolic neural networks with high-dimensional embeddings are necessary for enough model capacity. Moreover, current hyperbolic neural network layers are only designed for Poinaré ball model. To extend hyperbolic neural networks layers for Lorentz model can be an interesting future work.

References

[1] Octavian-Eugen Ganea, Gary Bécigneul, and Thomas Hofmann. Hyperbolic neural networks. *arXiv preprint arXiv:1805.09112*, 2018. 2

[2] Qi Liu, Maximilian Nickel, and Douwe Kiela. Hyperbolic graph neural networks. *arXiv preprint arXiv:1910.12892*, 2019. 7

[3] Weitang Liu, Xiaoyun Wang, John D Owens, and Yixuan Li. Energy-based out-of-distribution detection. *arXiv preprint arXiv:2010.03759*, 2020. 5

[4] Maximilian Nickel and Douwe Kiela. Poincar\'e embeddings for learning hierarchical representations. *arXiv preprint arXiv:1705.08039*, 2017. 6

[5] Maximillian Nickel and Douwe Kiela. Learning continuous hierarchies in the lorentz model of hyperbolic geometry. In *International Conference on Machine Learning*, pages 3779–3788. PMLR, 2018. 7

[6] Abraham A Ungar. Hyperbolic trigonometry and its application in the poincaré ball model of hyperbolic geometry. *Computers & Mathematics with Applications*, 41(1-2):135–147, 2001. 2

[7] Abraham A Ungar. *Analytic hyperbolic geometry: Mathematical foundations and applications*. World Scientific, 2005. 2

[8] Abraham Albert Ungar. A gyrovector space approach to hyperbolic geometry. *Synthesis Lectures on Mathematics and Statistics*, 1(1):1–194, 2008. 2