Deep Learning 1

2025-2026 – Pascal Mettes

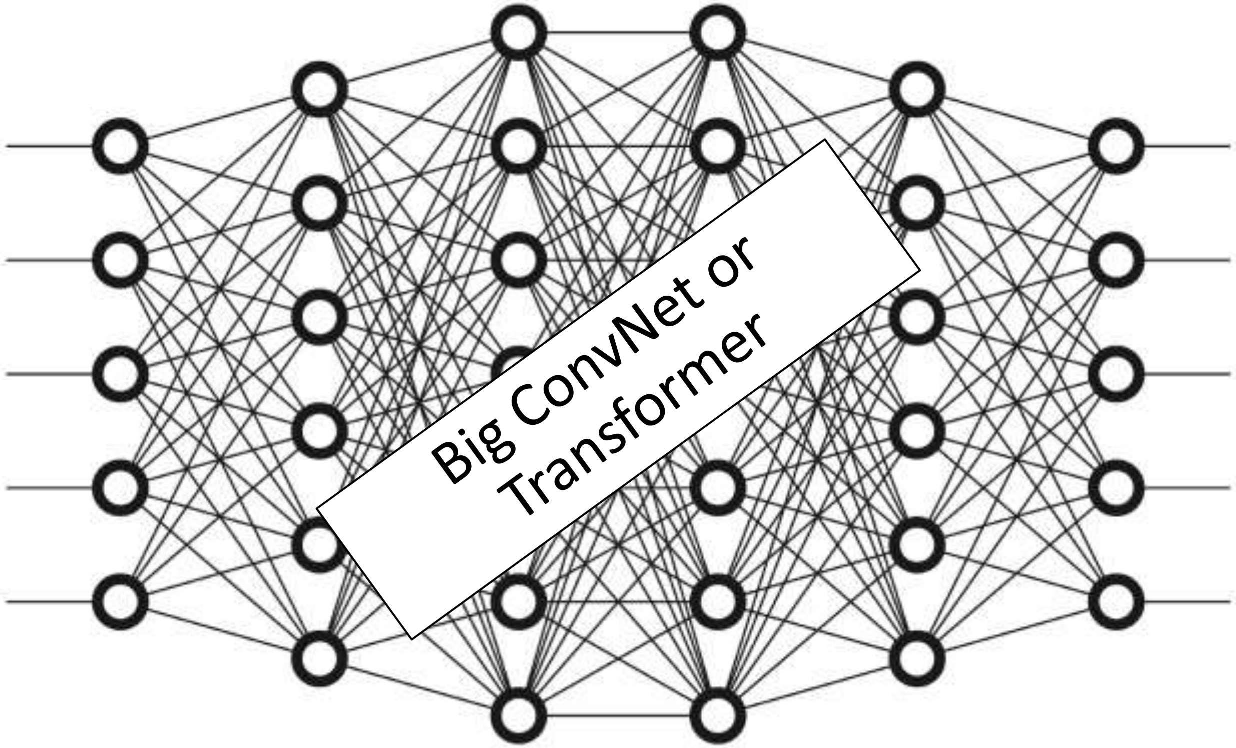
Lecture 12

Non-Euclidean deep learning

Previous lecture

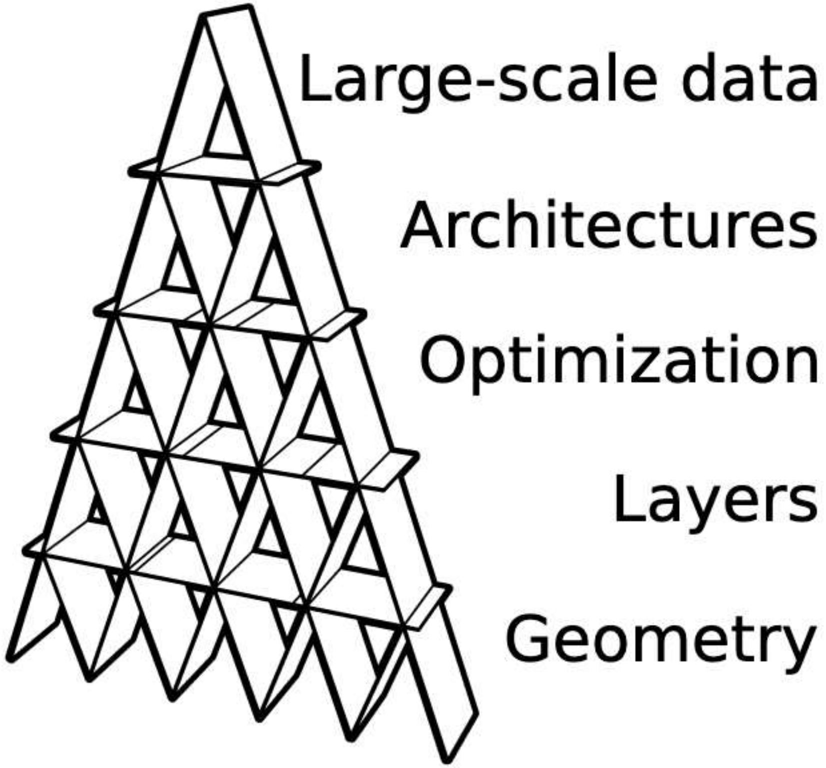
Lecture	Title	Lecture	Title
1	Intro and history of deep learning	2	AutoDiff
3	Deep learning optimization I	4	Deep learning optimization II
5	Convolutional deep learning	6	Attention-based deep learning
7	Graph deep learning	8	From supervised to unsupervised deep learning
9	Multi-modal deep learning	10	Generative deep learning
11	What doesn't work in deep learning	12	Non-Euclidean deep learning
13	Q&A	14	Deep learning for videos

Canonical deep learning



Person with dog

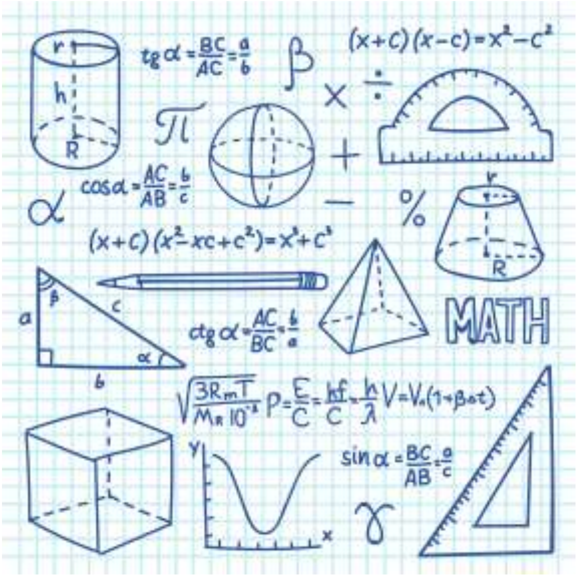
House of cards in deep learning



Our default choice of geometry in deep learning is Euclidean, but should it be?

Non-Euclidean geometry

We have a Euclidean bias in deep learning

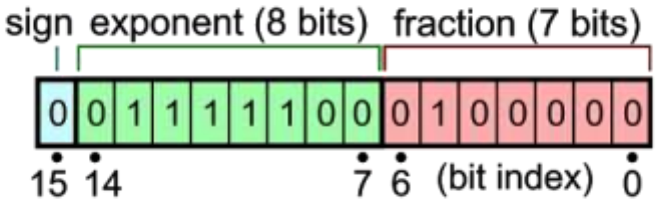


Our school curricula
are Euclidean

PyTorch

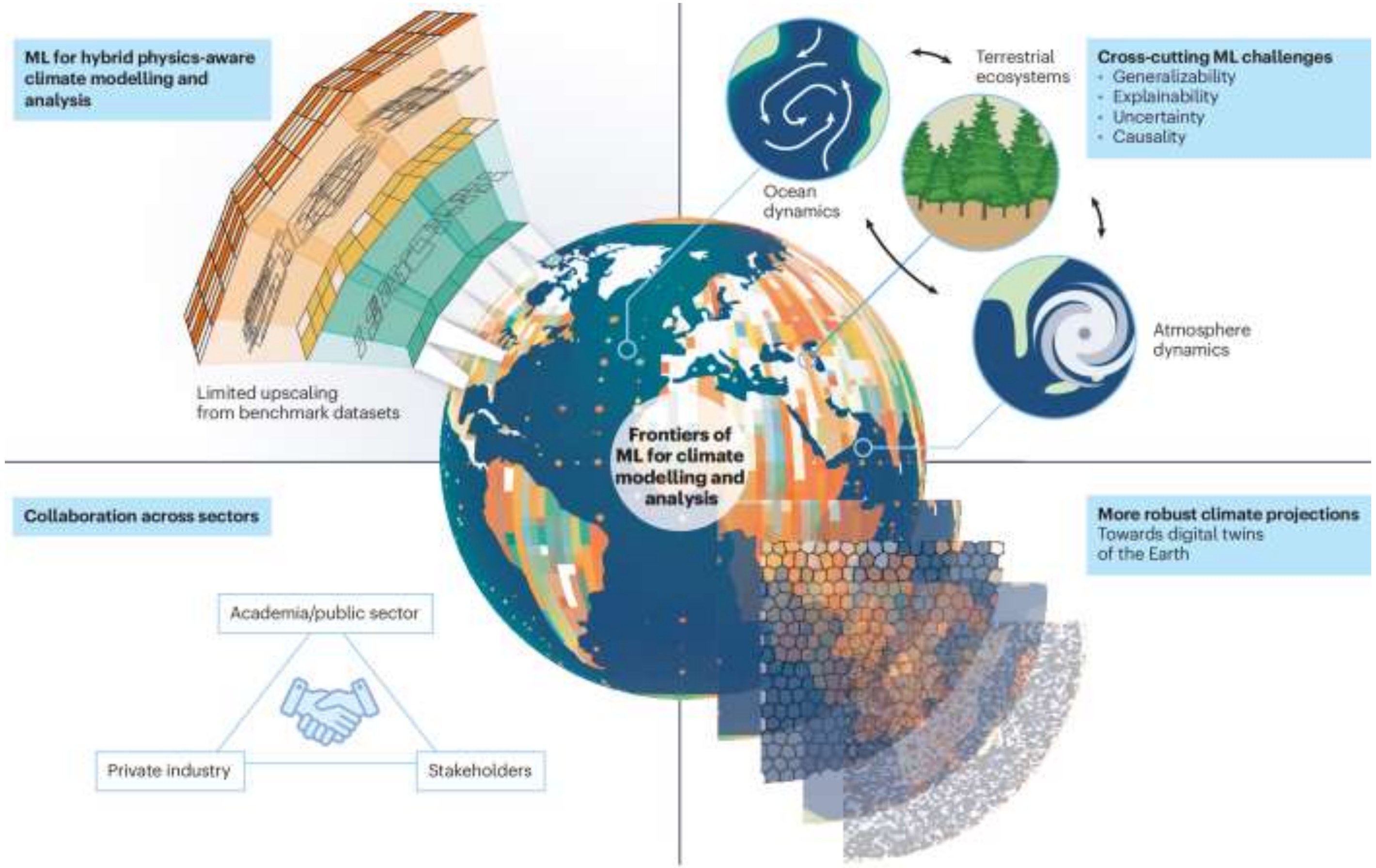
TensorFlow

Our deep learning tools
are Euclidean

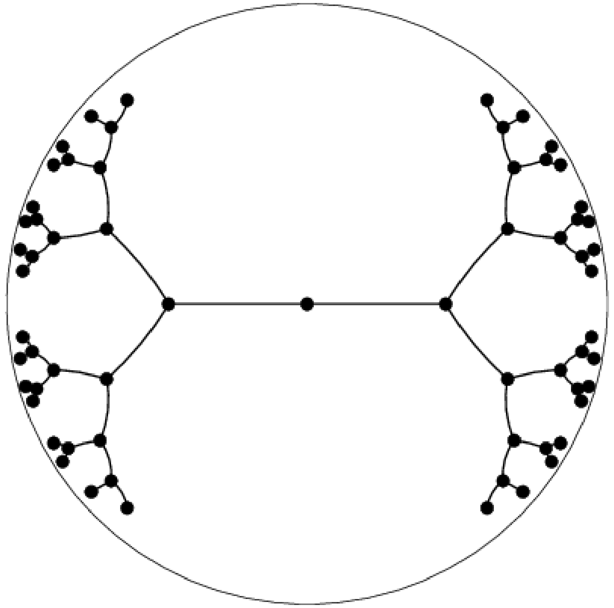


Our computers are built
for Euclidean space

But is Euclidean always the answer?

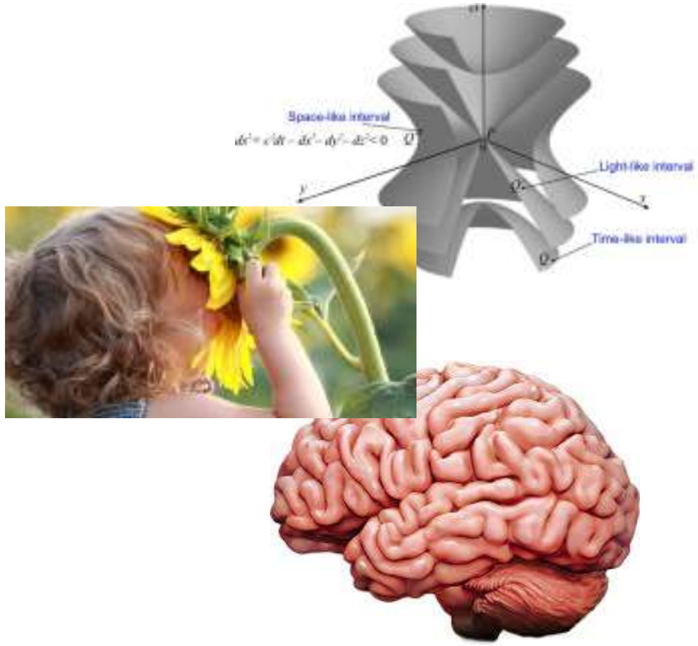


But is Euclidean always the answer?



Euclidean is not hierarchical

Euclidean is not compact



The world is not always Euclidean

Origins of non-Euclidean geometry

Euclid’s 5 postulates:

1. *A straight-line segment can be drawn joining any two points.*

2. Any straight-line segment can be extended indefinitely in a straight line.

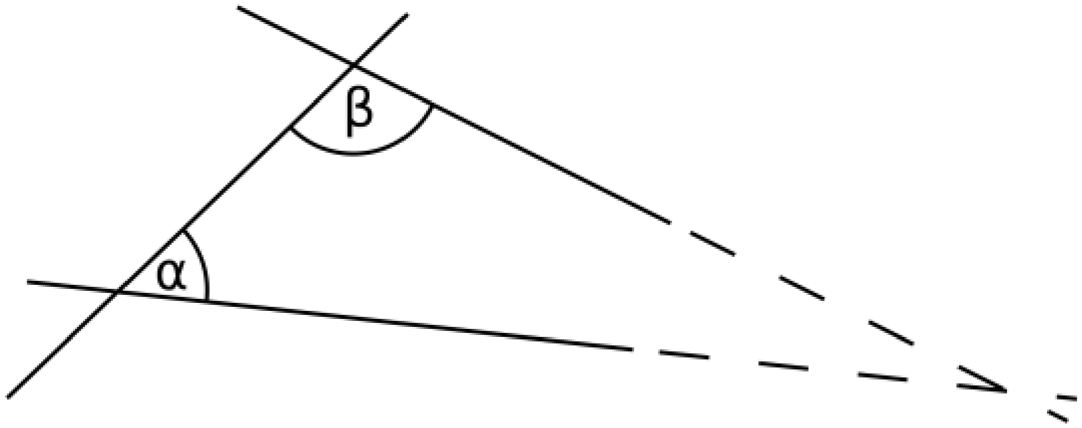
3. Given any straight lines segment, a circle can be drawn having the segment as radius and one endpoint as center.

4. All Right Angles are congruent.

5. If two lines are drawn which intersect a third in such a way that the sum of the inner angles on one side is less than two Right Angles, then the two lines inevitably must intersect each other on that side if extended far enough. This postulate is equivalent to what is known as the Parallel Postulate.

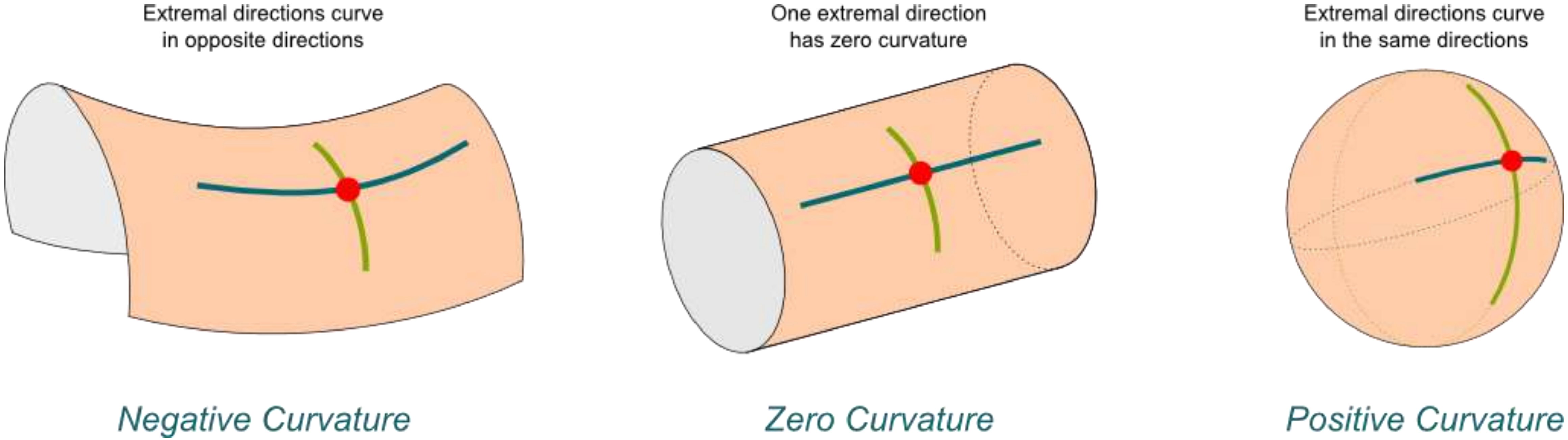
Origins of non-Euclidean geometry

Euclid’s 5th postulate:



Did Euclid make a mistake by making it a postulate? Shouldn’t it be a theorem?

Curving space



Hyperspherical deep learning

About spherical deep learning

Many real-world problems have data that lie on a sphere.

climate/weather modelling, world-wide patterns, 360 degree cameras, …

Even in standard neural networks, we often rely on spherical operations.

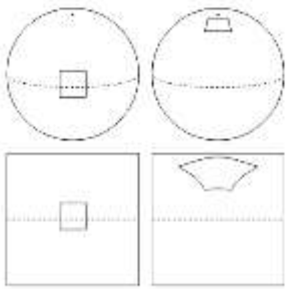
Cosine similarity!

L2 data normalization

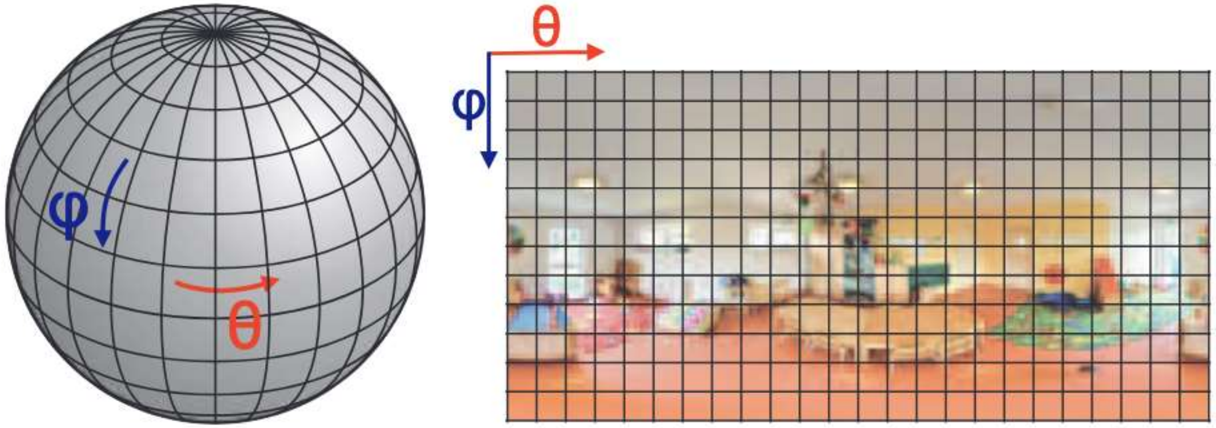
Classifiers

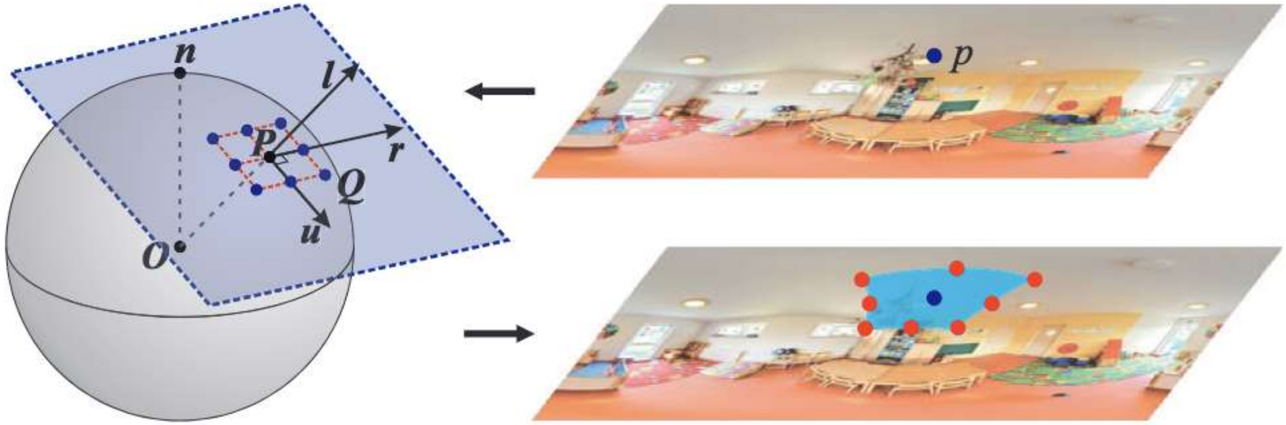
In standard networks, a spherical perspective can bring new insights.

Deep learning on spherical inputs



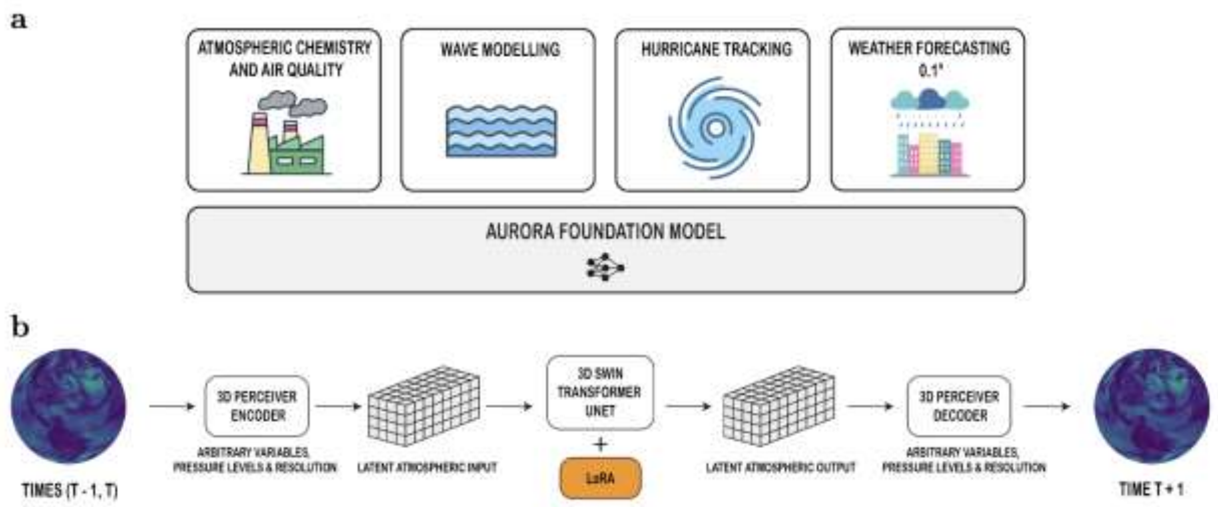
Cohen et al. (2018)

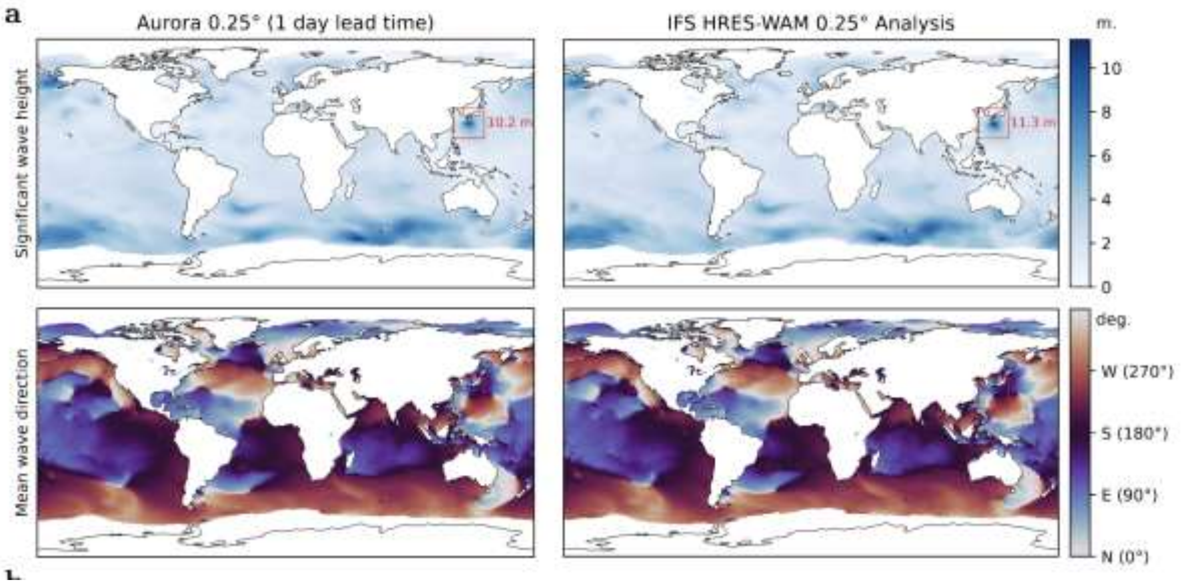




Zhao et al. (2018)

Yet spherical distortion can also be ignored



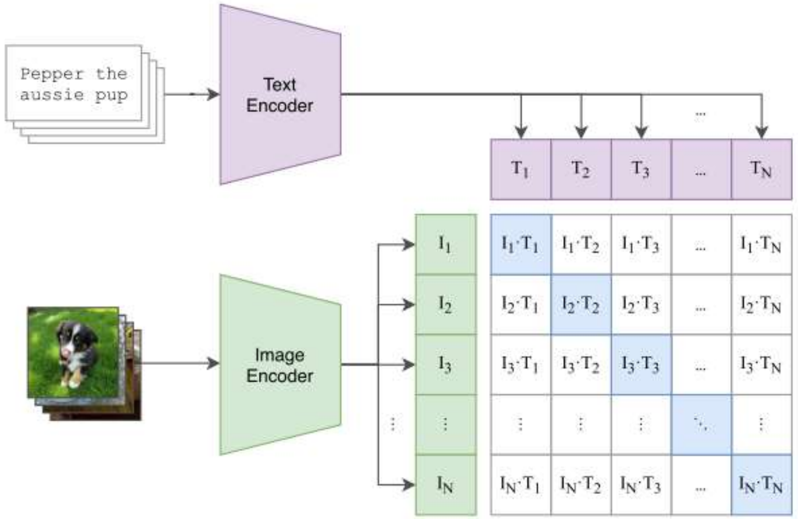


A Foundation Model for the Earth System – Bodnar et al. (2024)

Spherical losses are everywhere in deep learning

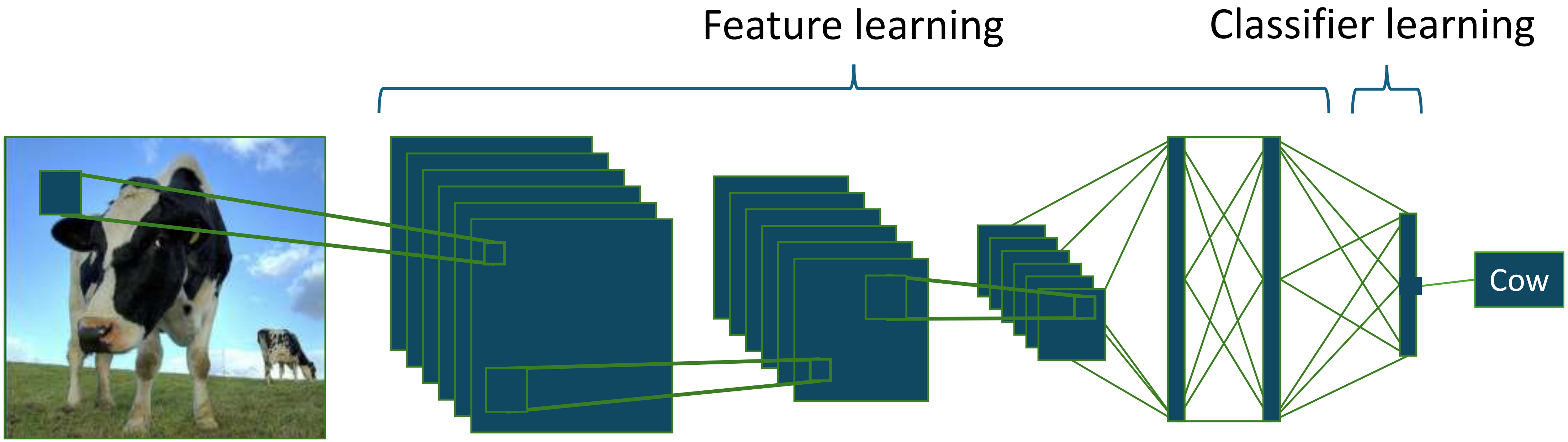
“On the Surprising Behavior of Distance Metrics in High Dimensional Space” – Aggarwal et al. (2001)

“What is the nearest neighbor in high-dimensional spaces?” – Hinneburg et al. (2000)




```
# extract feature representations of each modality
I_f = image_encoder(I) #[n, d_i]
T_f = text_encoder(T)  #[n, d_t]

# joint multimodal embedding [n, d_e]
I_e = l2_normalize(np.dot(I_f, W_i), axis=1)
T_e = l2_normalize(np.dot(T_f, W_t), axis=1)

# scaled pairwise cosine similarities [n, n]
logits = np.dot(I_e, T_e.T) * np.exp(t)

# symmetric loss function
labels = np.arange(n)
loss_i = cross_entropy_loss(logits, labels, axis=0)
loss_t = cross_entropy_loss(logits, labels, axis=1)
loss   = (loss_i + loss_t)/2
```

Classifiers in deep networks are spherical



Recall this decoupling from before.

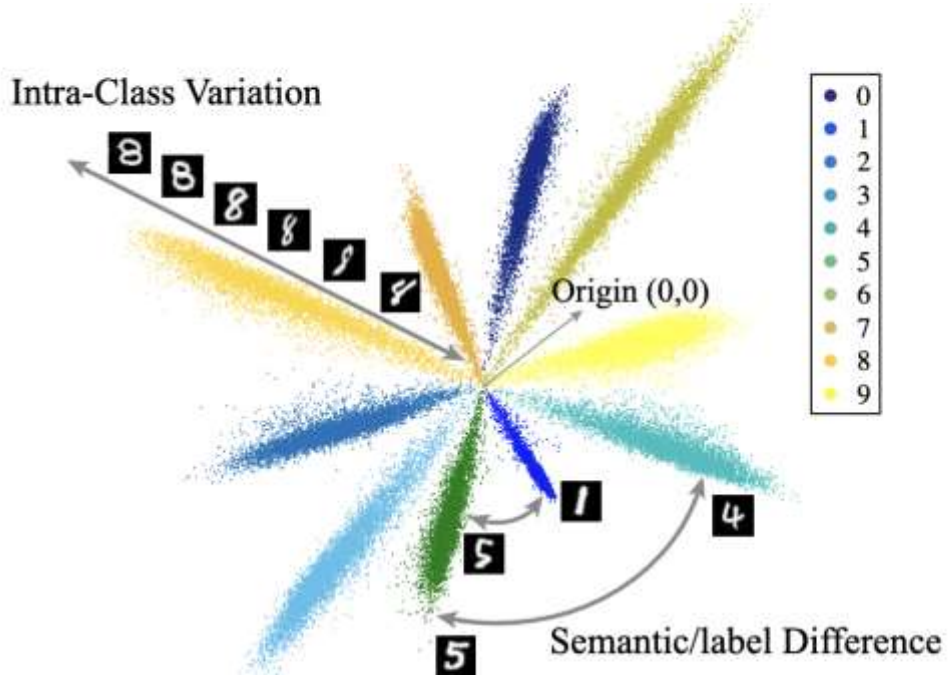
If we think of classifiers as angularly separated, what makes an optimal classifier?

Decoupled Networks – Liu et al. (2018)

Each class-specific classifier is a vector in the last layer of the network.

Similarity to sample is given by the dot-product.

Hence: high similarity of classifier and sample point in the same direction!



Maximum Class Separation – Kasarla et al. (2022)

If classes are separated by angles, there is a theoretical optimal last layer.

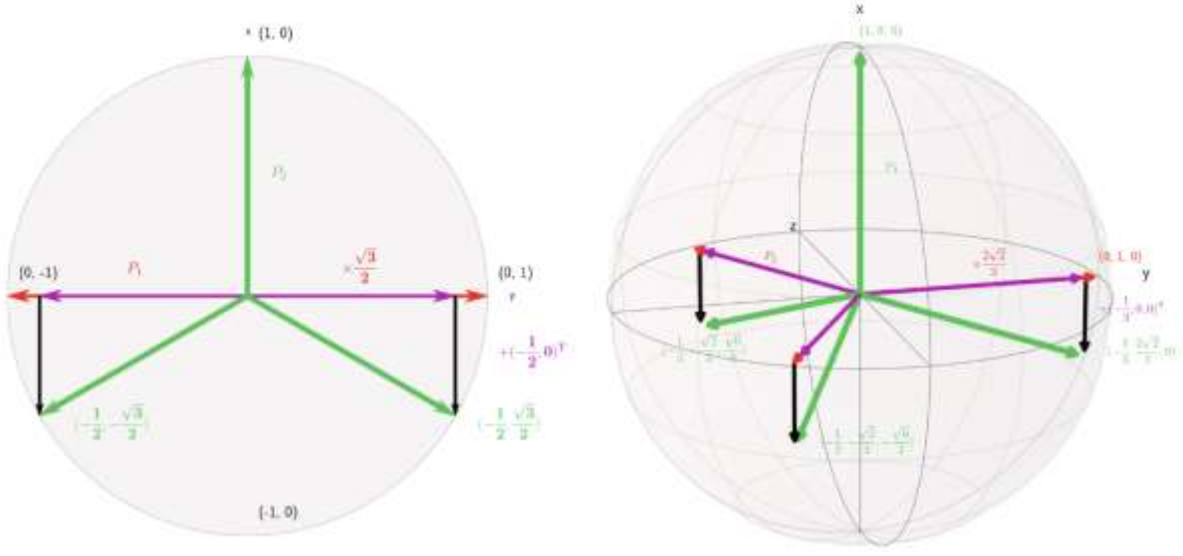
Separate all classes uniformly on the hypersphere.

General problem known as Tammes problem.

Specifically, for N classes in N-1 space, there is a solution:

$$P_1 = \begin{pmatrix} 1 & -1 \end{pmatrix} \in \mathbb{R}^{1\times 2}$$

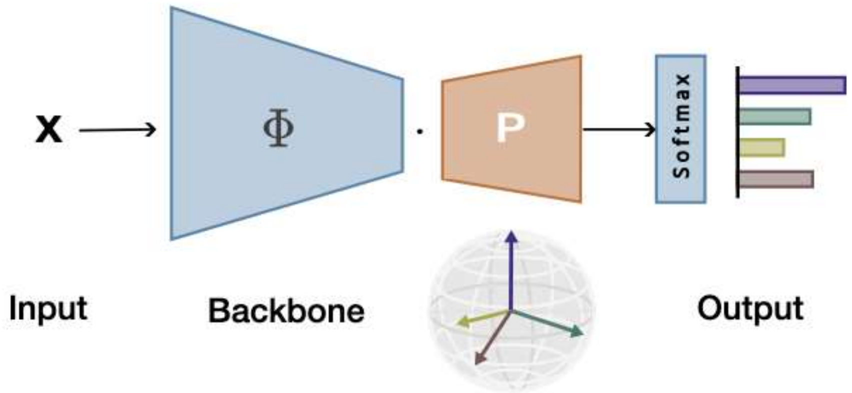
$$P_k = \begin{pmatrix} 1 & -\frac{1}{k}\mathbf{1}^T \\ \mathbf{0} & \sqrt{1-\frac{1}{k^2}}\, P_{k-1} \end{pmatrix} \in \mathbb{R}^{k\times(k+1)}$$



(a) Recursive update from 2 to 3 classes.

(b) Recursive update from 3 to 4 classes.

Maximum Class Separation – Kasarla et al. (2022)



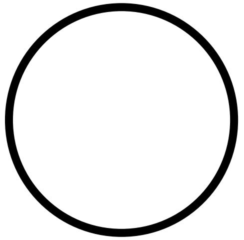
	Resnet-50				Resnet-152			
	top 1		top 5		top 1		top 5	
	Base	+ Ours	Base	+ Ours	Base	+ Ours	Base	+ Ours
Imagenet	73.2	**74.8**	92.4	**94.9**	77.9	**78.5**	94.3	**95.1**
Imagenet-LT	43.8	**47.3**	70.4	**73.6**	48.3	**49.7**	73.9	**74.8**

	Imbalance factor 0.1	0.02	0.01		Imbalance factor 0.1	0.02	0.01
LDAM-SGD	55.05	43.85	39.87	MiSLAS (stage 1)	58.36	44.69	40.29
+ This paper	**57.72**	**45.14**	**42.02**	+ This paper	**59.63**	**45.65**	**40.56**
	+2.67	+1.29	+2.20		+1.27	+0.96	+0.27
LDAM-DRW	57.45	47.56	42.37	MiSLAS (stage 2)	61.93	52.53	48.00
+ This paper	**58.37**	**48.02**	**43.19**	+ This paper	**63.52**	**53.36**	**48.42**
	+0.92	+0.46	+0.82		+1.59	+0.83	+0.42

	CIFAR-100					CIFAR-10				
	-	0.2	0.1	0.02	0.01	-	0.2	0.1	0.02	0.01
ConvNet	56.70	45.97	40.34	27.35	16.59	86.68	79.47	73.90	51.40	43.67
+ This paper	**57.05**	**46.59**	**40.44**	**28.27**	**18.40**	**86.76**	**79.63**	**75.88**	**55.25**	**48.05**
	+0.35	+0.62	+0.10	+0.92	+1.81	+0.08	+0.16	+1.98	+3.85	+4.38
ResNet-32	75.77	65.74	58.98	42.71	35.02	94.63	88.17	83.10	68.64	56.98
+ This paper	**76.54**	**66.01**	**60.54**	**45.12**	**38.85**	**95.09**	**91.42**	**88.16**	**77.02**	**69.70**
	+0.77	+0.27	+1.56	+2.41	+3.83	+0.46	+3.25	+5.06	+8.38	+12.72

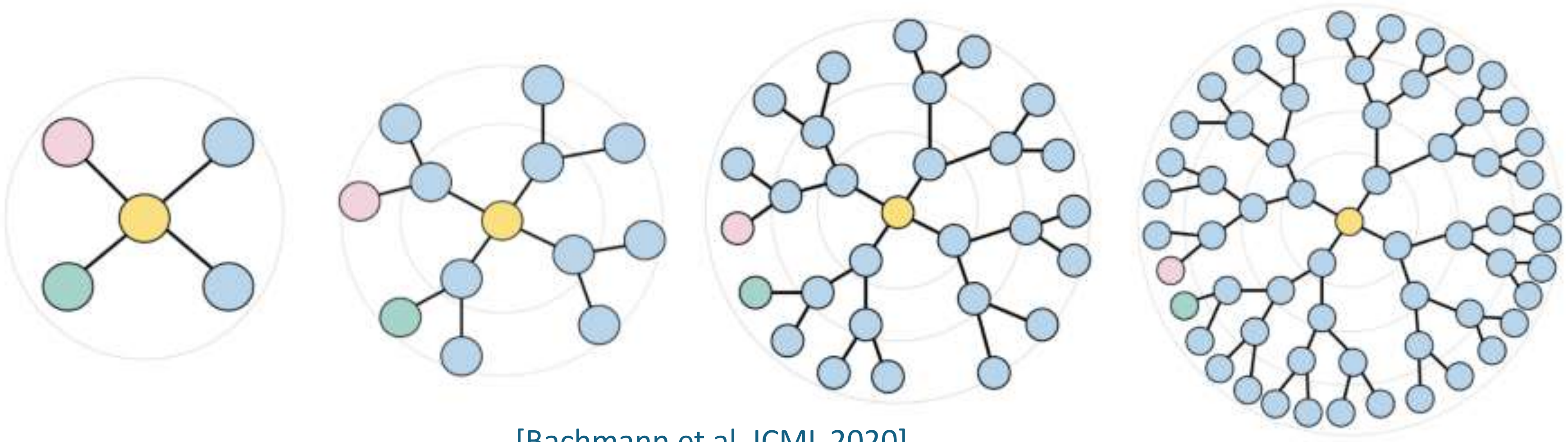
Hyperbolic deep learning

The geometry of hierarchies



$v = \pi r^2$

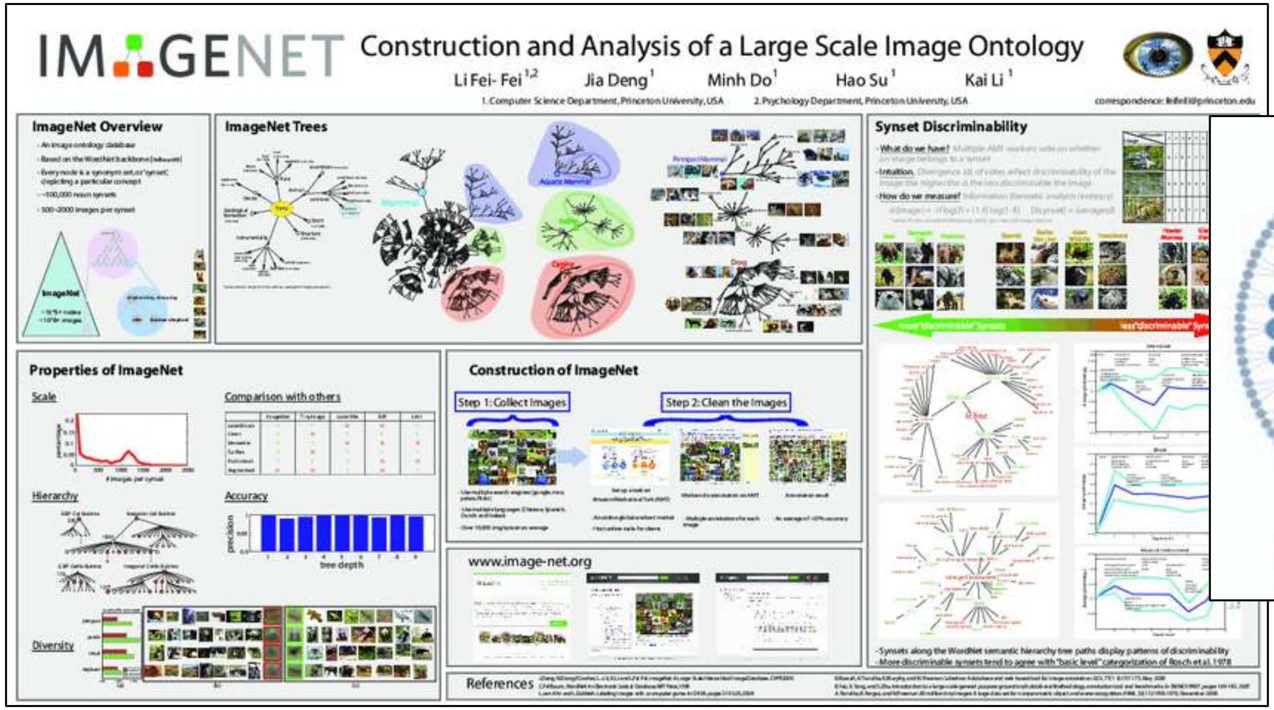
Euclidean space and hierarchies are a mismatch: linear vs. exponential growth.



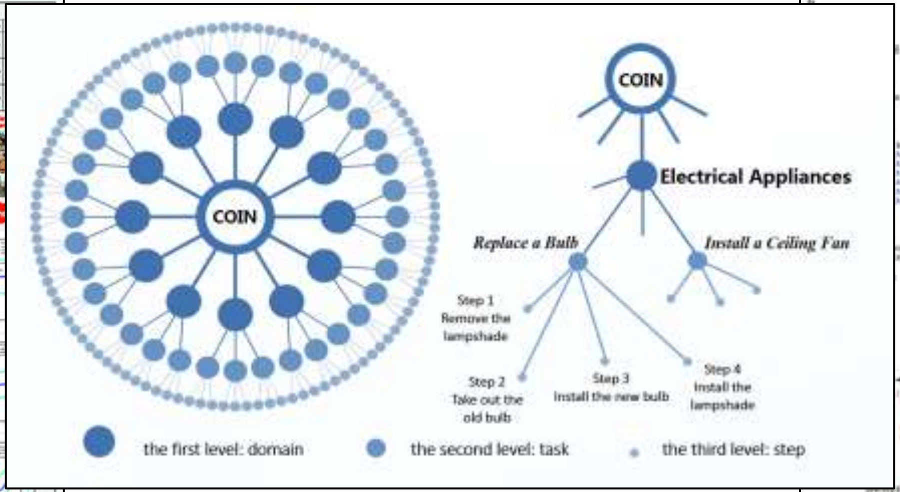
[Bachmann et al. ICML 2020]

What we need is a hierarchical geometry for representation learning!

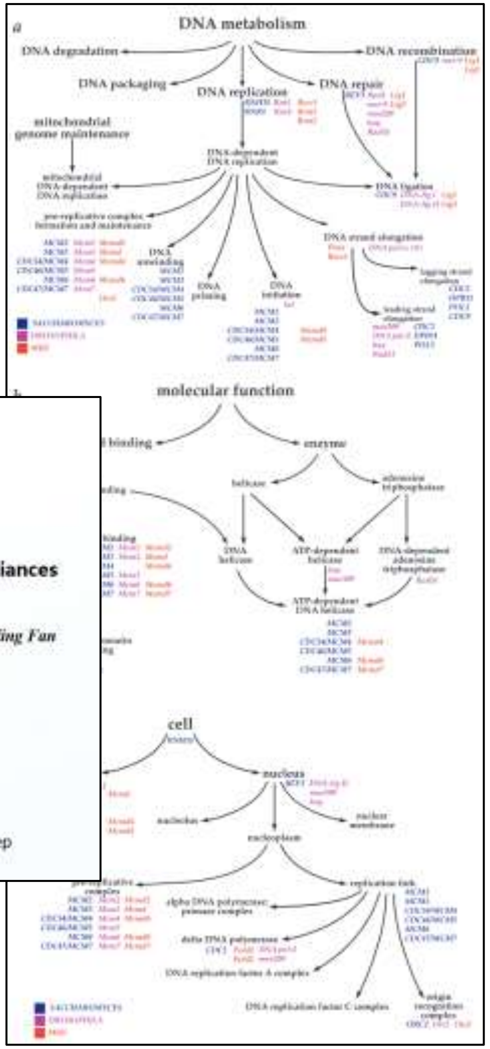
The importance of hierarchies



[Li et al. 2009]

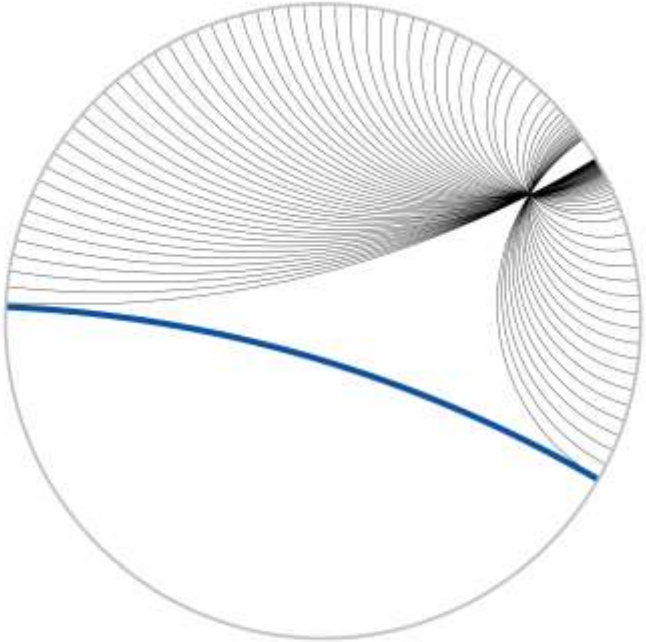


[Tang et al. CVPR 2019]

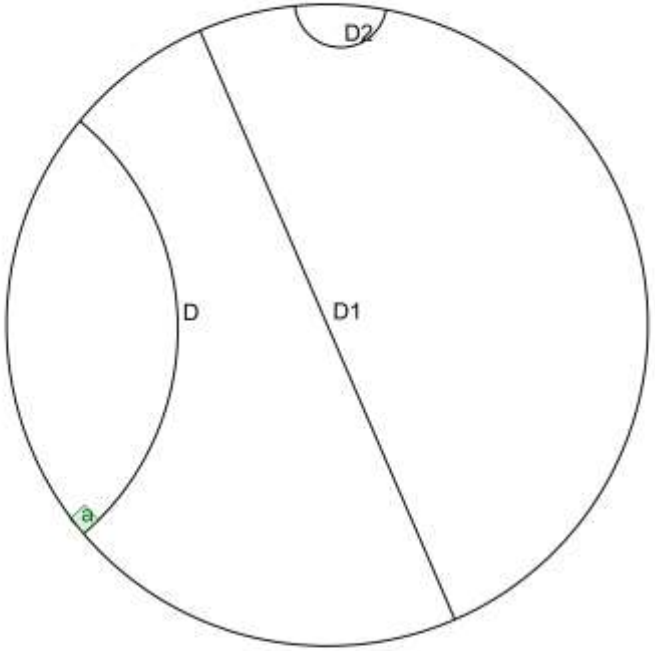


[Ashburner et al. Nature 2000]

Hierarchies allow us to look beyond samples and their individual labels.

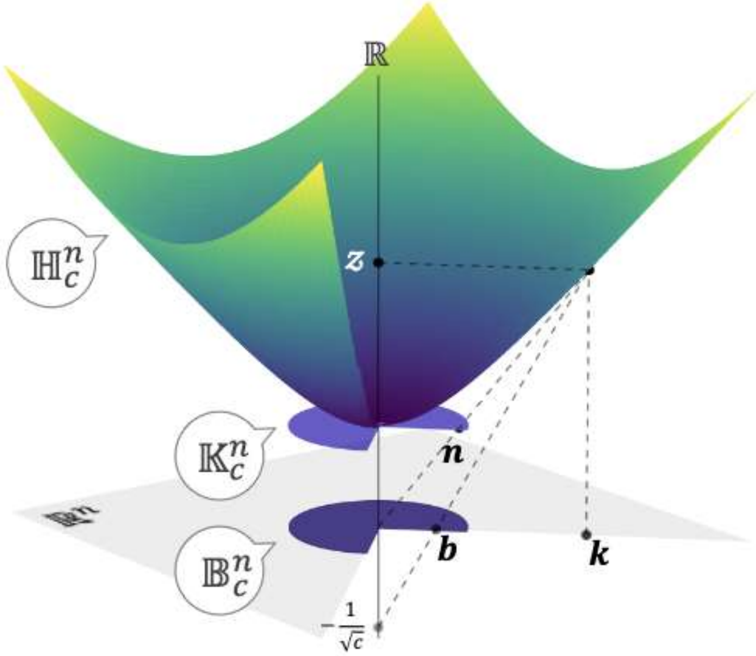


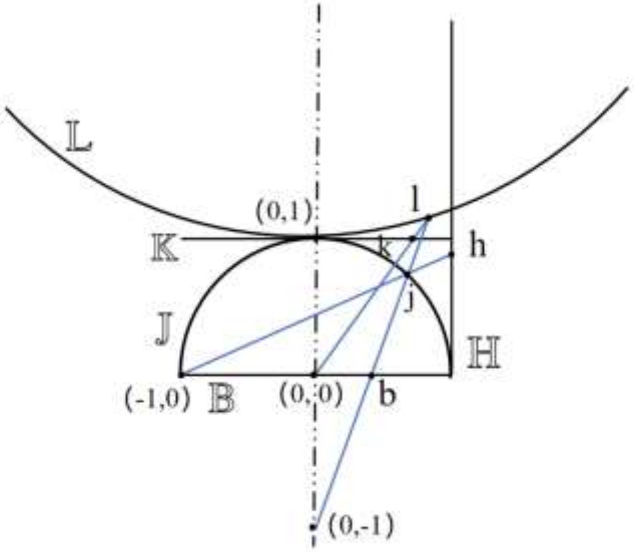
Poincaré ball model



Models of hyperbolic geometry

To perform numerical operations, we need to operate in a model of hyperbolic geometry.





Multiple isometric models exist, with different pros and cons for numerical complexity, stability, and visualization prowess.

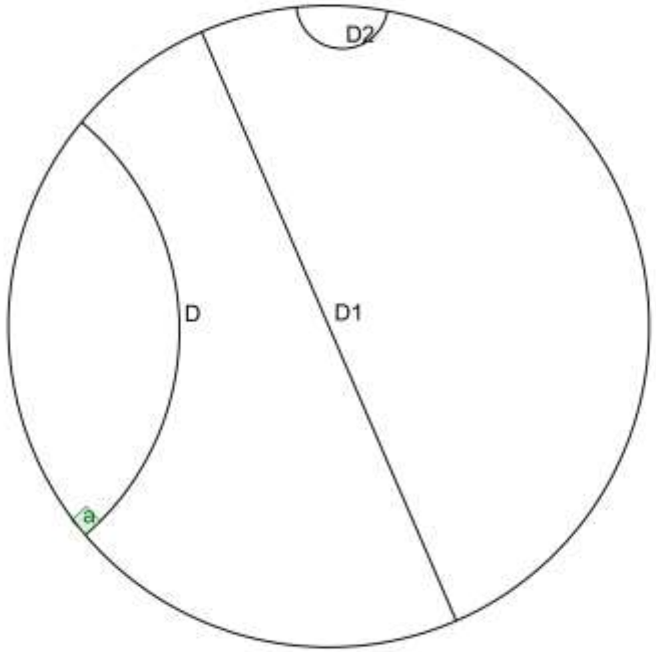
Numerical operation in Poincaré model

Points inside unit ball

$$\mathbb{D}^n = \{x \in \mathbb{R}^n : \|x\| < 1\}$$

Tensor metric

$$g_x^{\mathbb{D}} = \lambda_x^2 g^E, \quad \text{where } \lambda_x := \frac{2}{1 - \|x\|^2}$$



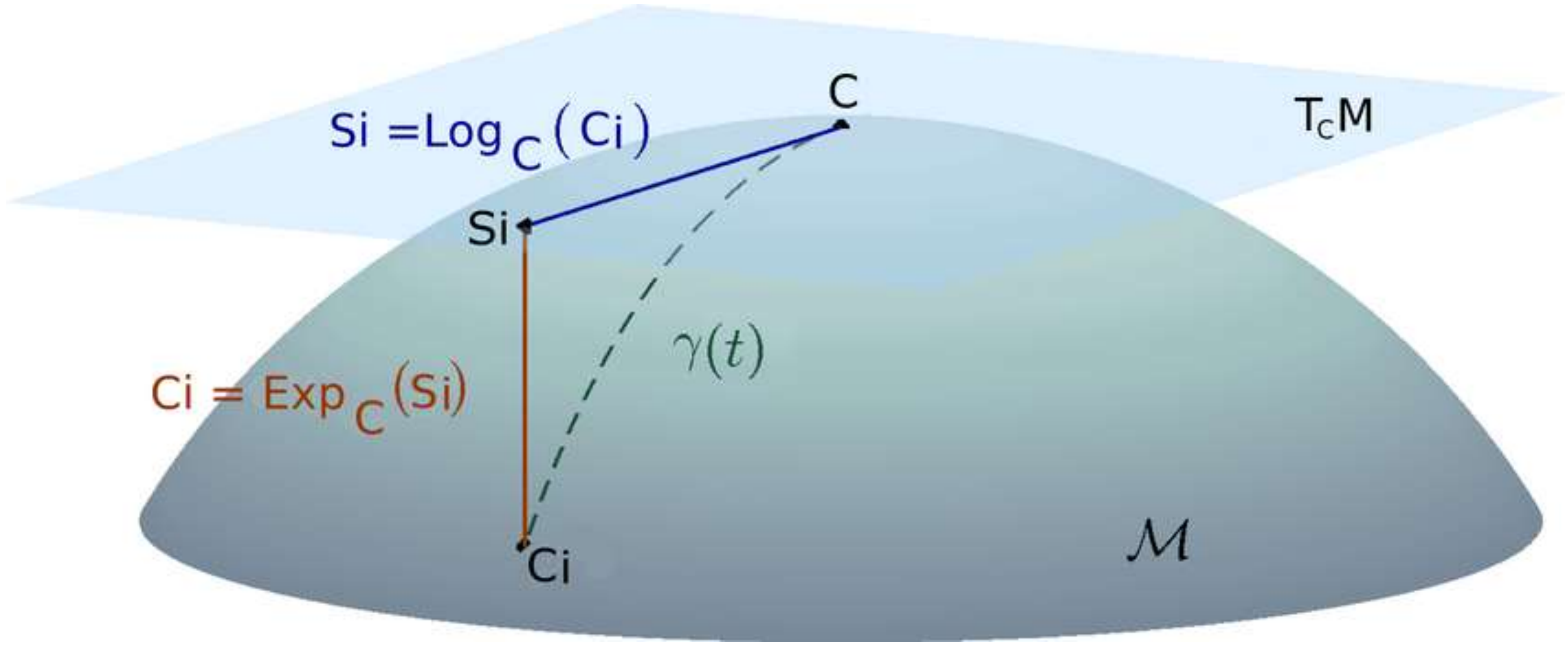
Distance between two points:

$$d_{\mathbb{D}}(x, y) = \cosh^{-1}\left(1 + 2\frac{\|x - y\|^2}{(1 - \|x\|^2)(1 - \|y\|^2)}\right)$$

Möbius addition:

$$x \oplus_c y := \frac{(1 + 2c\langle x, y\rangle + c\|y\|^2)x + (1 - c\|x\|^2)y}{1 + 2c\langle x, y\rangle + c^2\|x\|^2\|y\|^2}.$$

From tangent space to Poincaré ball (and back)



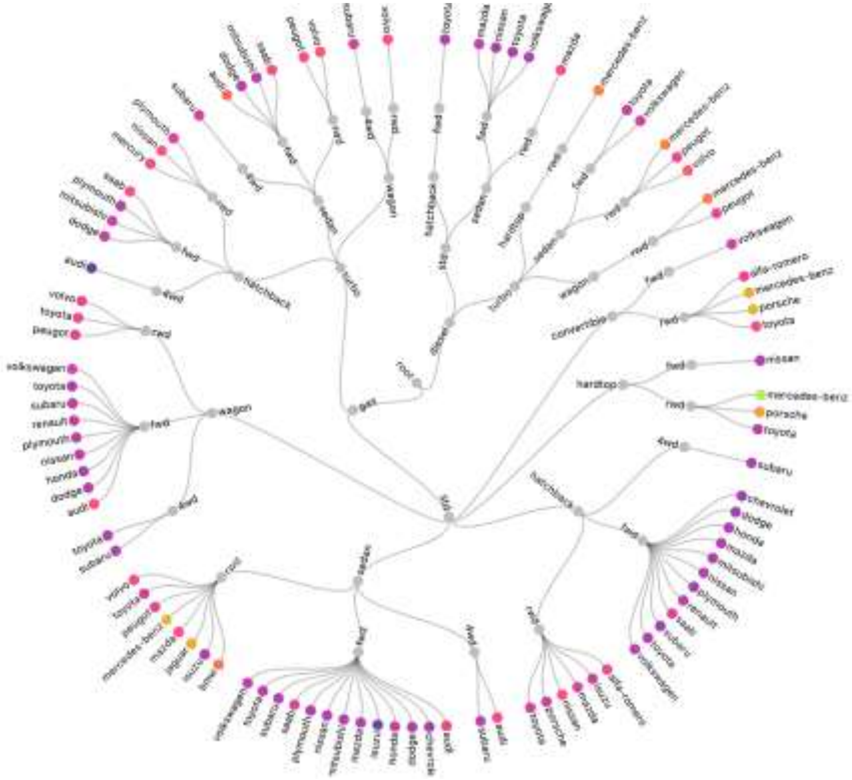
$$\exp_{\mathbf{0}}^{c}(v) = \tanh(\sqrt{c}\|v\|)\frac{v}{\sqrt{c}\|v\|}$$

$$\log_{\mathbf{0}}^{c}(y) = \tanh^{-1}(\sqrt{c}\|y\|)\frac{y}{\sqrt{c}\|y\|}$$

First hyperbolic success: Poincaré Embeddings

[Nickel and Kiela. NeurIPS 2017]

Embed nodes as hyperbolic points and optimize with contrastive learning.



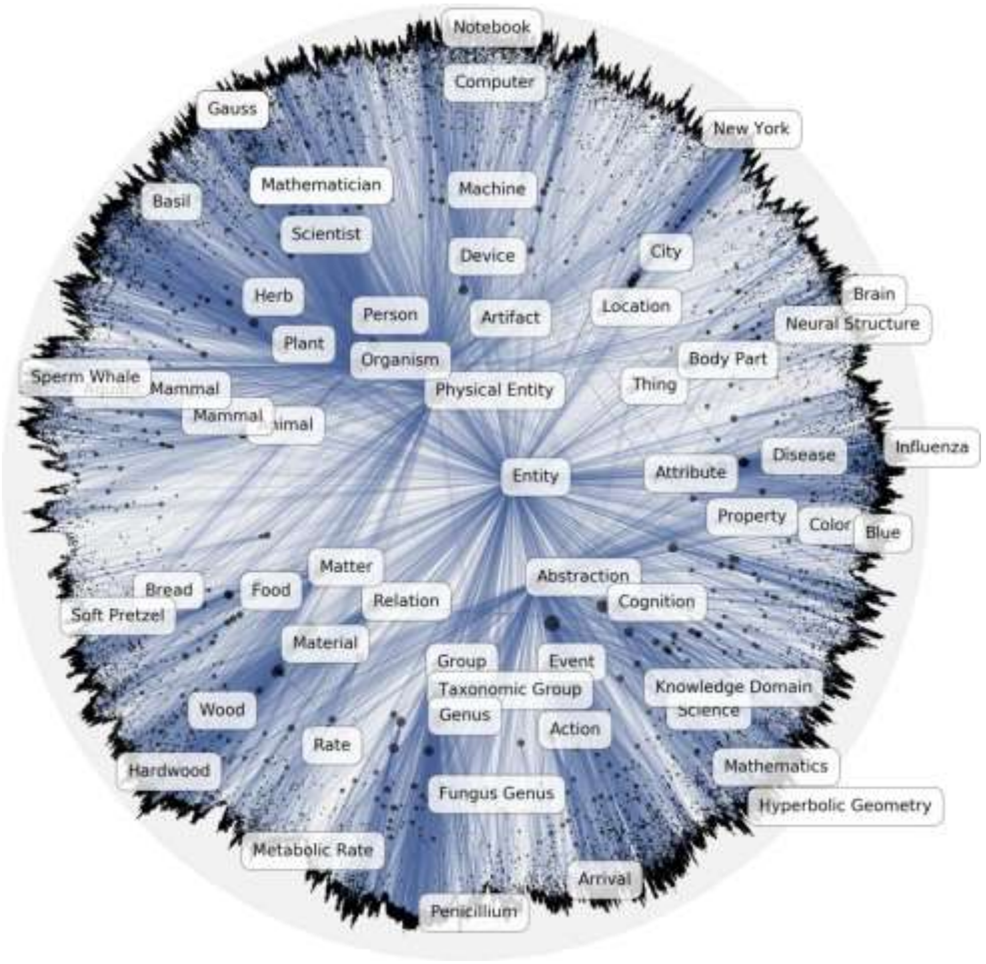
$$\mathcal{L}(\Theta) = \sum_{(u,v)\in\mathcal{D}} \log \frac{e^{-d(\boldsymbol{u},\boldsymbol{v})}}{\sum_{\boldsymbol{v}'\in\mathcal{N}(u)} e^{-d(\boldsymbol{u},\boldsymbol{v}')}},$$

Parent-child links

Non parent-child links

Hyperbolic distance

Poincaré Embeddings



			Dimensionality					
			5	10	20	50	100	200
WORDNET Reconstruction	Euclidean	Rank	3542.3	2286.9	1685.9	1281.7	1187.3	1157.3
		MAP	0.024	0.059	0.087	0.140	0.162	0.168
	Translational	Rank	205.9	179.4	95.3	92.8	92.7	91.0
		MAP	0.517	0.503	0.563	0.566	0.562	0.565
	Poincaré	Rank	4.9	4.02	3.84	3.98	3.9	**3.83**
		MAP	0.823	0.851	0.855	0.86	0.857	**0.87**
WORDNET Link Pred.	Euclidean	Rank	3311.1	2199.5	952.3	351.4	190.7	81.5
		MAP	0.024	0.059	0.176	0.286	0.428	0.490
	Translational	Rank	65.7	56.6	52.1	47.2	43.2	40.4
		MAP	0.545	0.554	0.554	0.56	0.562	0.559
	Poincaré	Rank	5.7	**4.3**	4.9	4.6	4.6	4.6
		MAP	0.825	0.852	0.861	**0.863**	0.856	0.855

Hyperbolic Entailment Cones

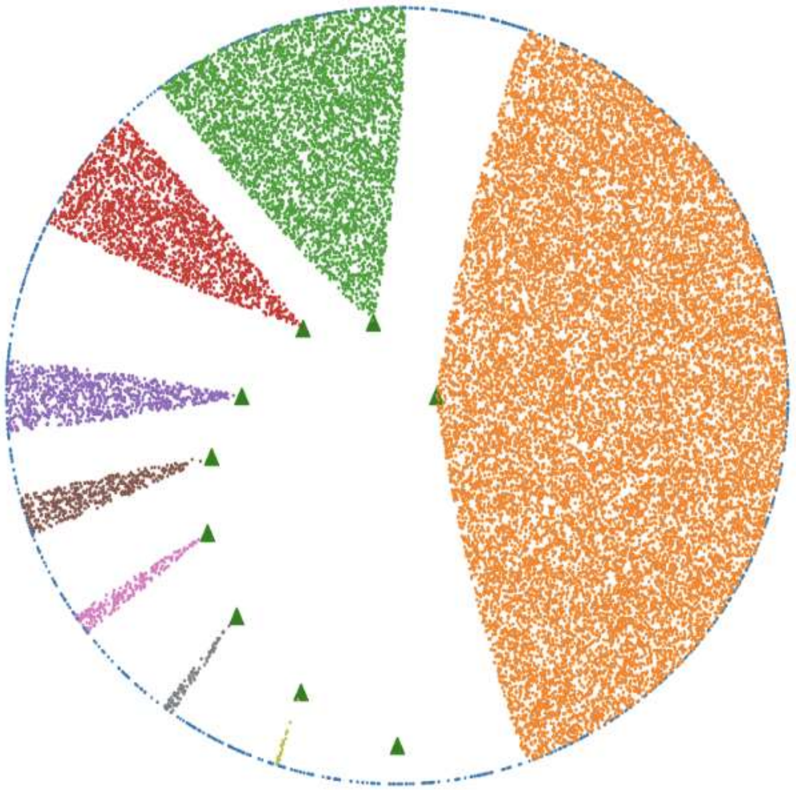
[Ganea et al. ICML 2018]

Pairwise contrastive learning has trouble enforcing hierarchical depth.

They propose to view points as cones of entailment.

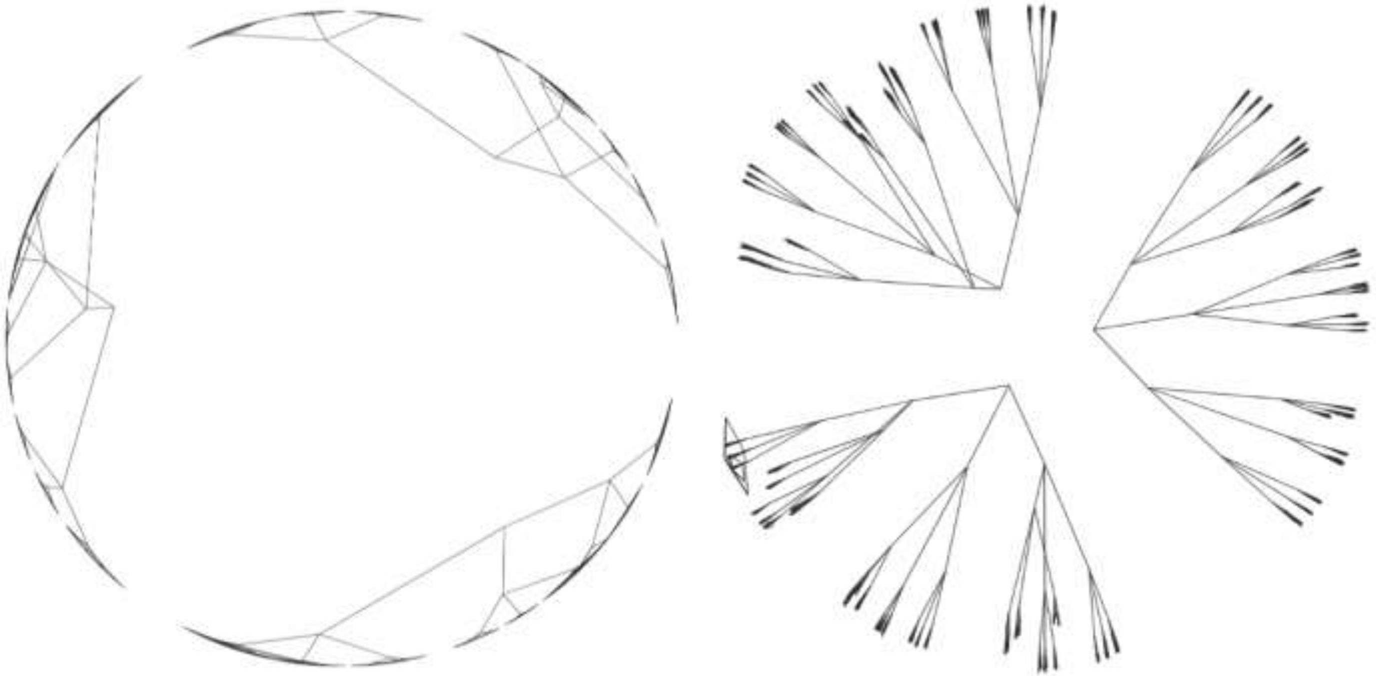
Required properties:

1. Axial symmetry
2. Rotation invariance
3. Aperture of cone is continuous function
4. Nested angular cones preserve partial order



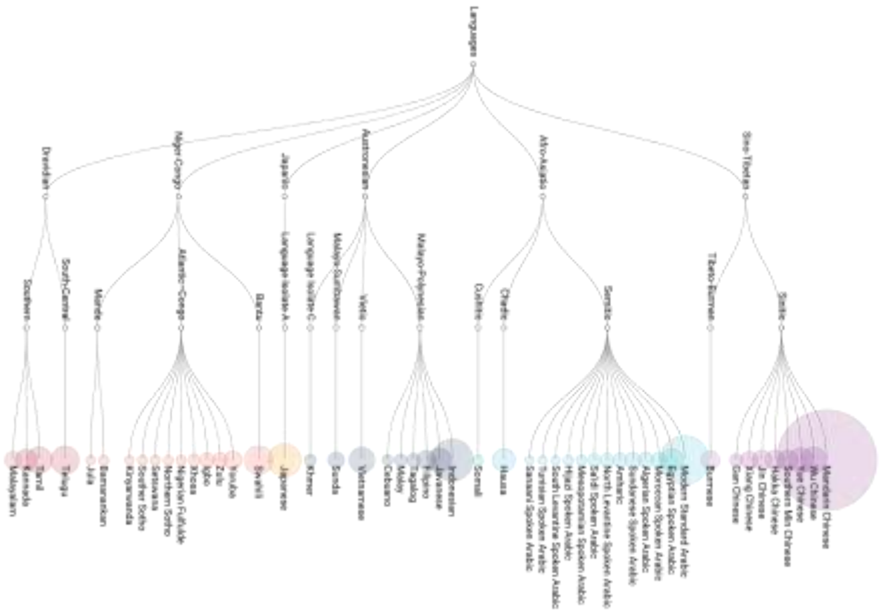
Hyperbolic Entailment Cones

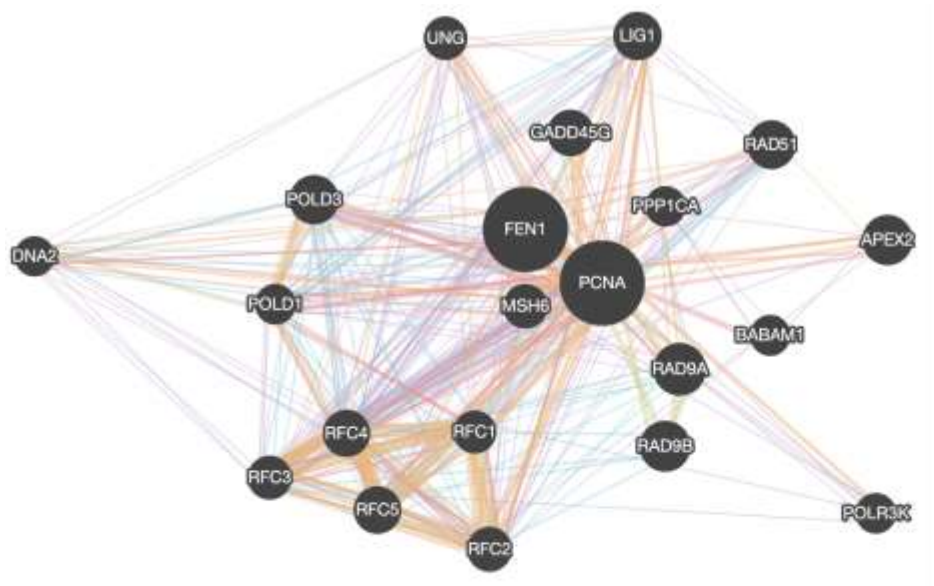
[Ganea et al. ICML 2018]



Poincaré Embeddings (left) vs Hyperbolic Entailment Cones (right)

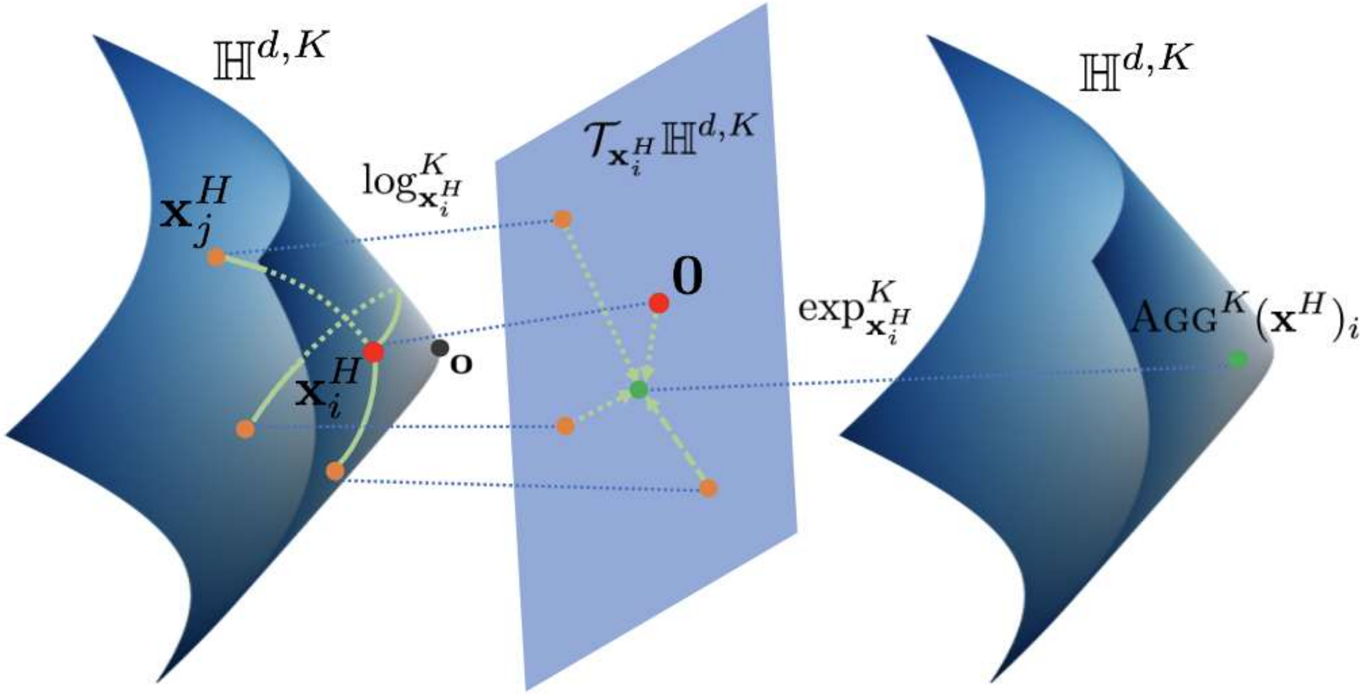
From hierarchies to graphs





Nodes and edges	Nodes and edges
Embed to preserve edge distances	Embed for node/edge/graph inference
One layer	Multiple layers

First generation hyperbolic graph networks



Project hyperbolic nodes to Euclidean space, do normal graph layer, project back.

[Chami et al., NeurIPS 2019, Liu et al., NeurIPS 2019]

First generation hyperbolic graph networks

Standard graph layer

$$\mathbf{h}_u^{k+1} = \sigma \left(\sum_{v \in \mathcal{I}(u)} \tilde{\mathbf{A}}_{uv} \mathbf{W}^k \mathbf{h}_v^k \right)$$

$$\tilde{\mathbf{A}} = \mathbf{D}^{-\frac{1}{2}} (\mathbf{A} + \mathbf{I}) \mathbf{D}^{-\frac{1}{2}}$$

Normalized affinity matrix

Weights to learn

Current node embeddings

Hyperbolic graph layer

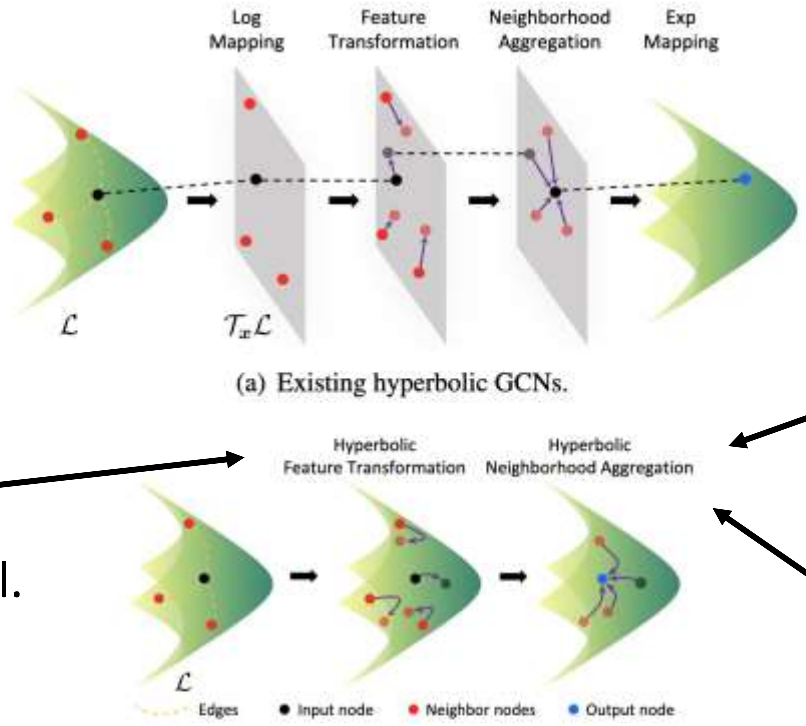
$$\mathbf{h}_u^{k+1} = \sigma \left(\exp_{\mathbf{x}'} (\sum_{v \in \mathcal{I}(u)} \tilde{\mathbf{A}}_{uv} \mathbf{W}^k \log_{\mathbf{x}'} (\mathbf{h}_v^k)) \right)$$

Projection steps

$$\exp_{\mathbf{x}}(\mathbf{v}) = \mathbf{x} + \mathbf{v} \qquad \log_{\mathbf{x}}(\mathbf{y}) = \mathbf{y} - \mathbf{x}$$

Second generation hyperbolic graph networks

Perform transformations also in hyperbolic space.



Aggregation in Klein model.

Transformations in Lorentz model.

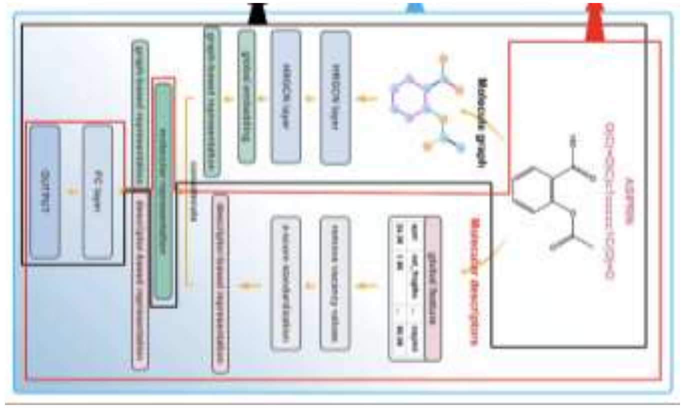
Non-linearities in Poincaré ball model.

[Dai et al. CVPR 2021]

Applications of hyperbolic graph networks

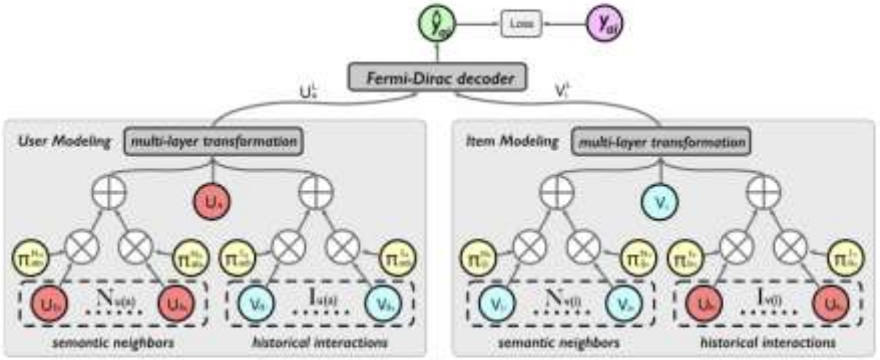
Molecules

[Wu et al. BiB 2021]



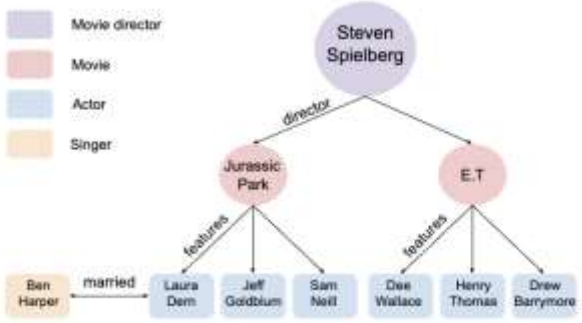
Recommendation systems

[Li et al. TKDE 2023]



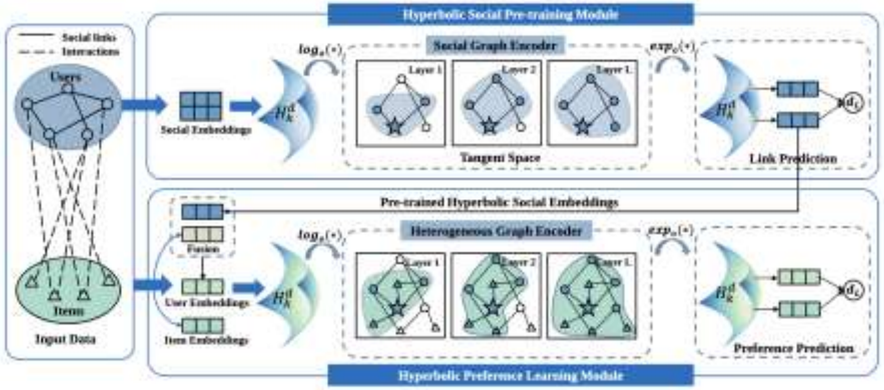
Knowledge graphs

[Chami et al. 2020]



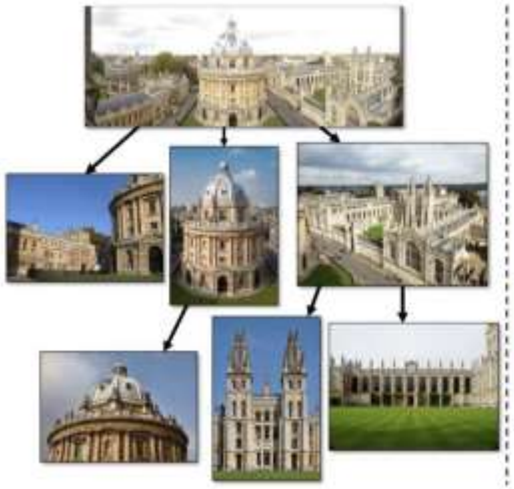
Social networks

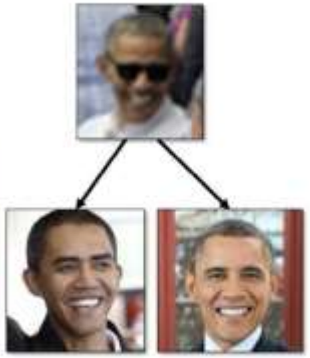
[Yang et al. TKDE 2023]

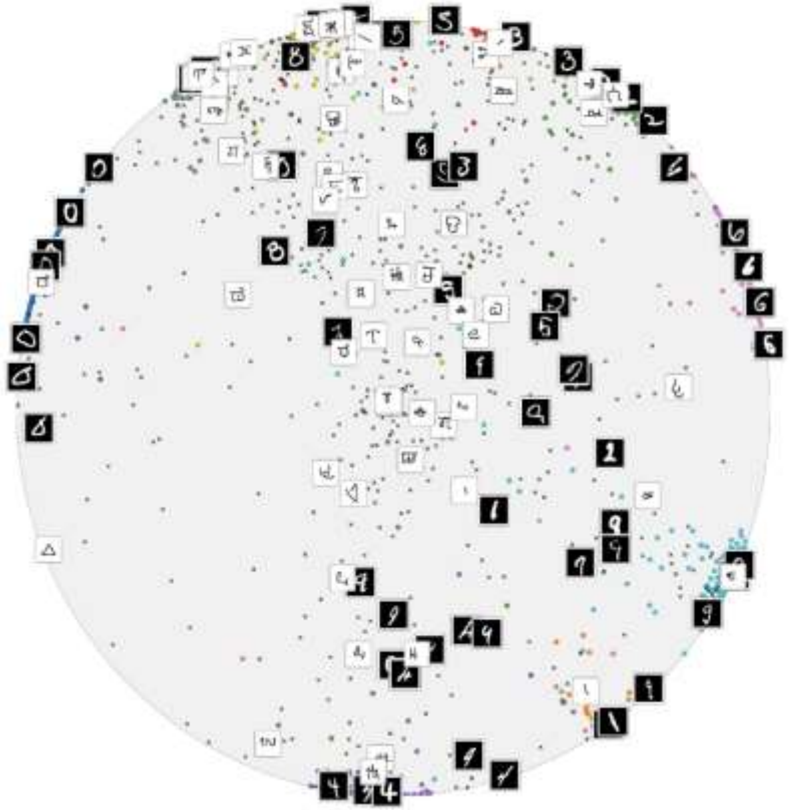


Hyperbolic Image Embeddings

[Khrulkov et al. CVPR 2020]





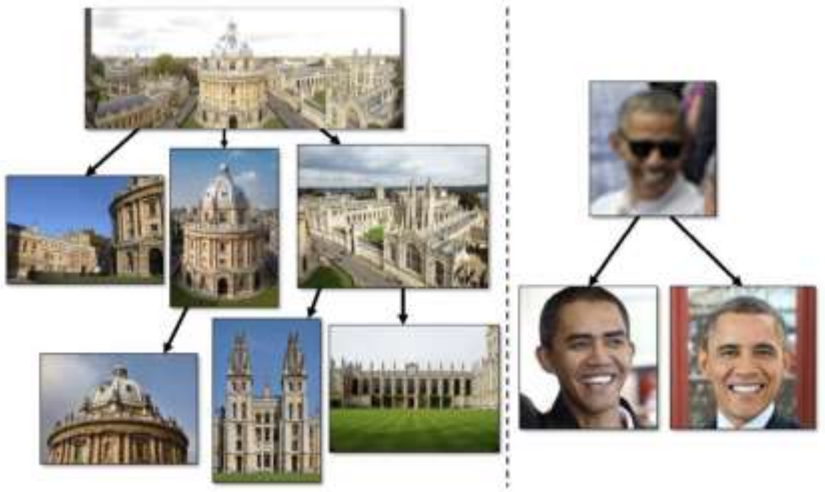


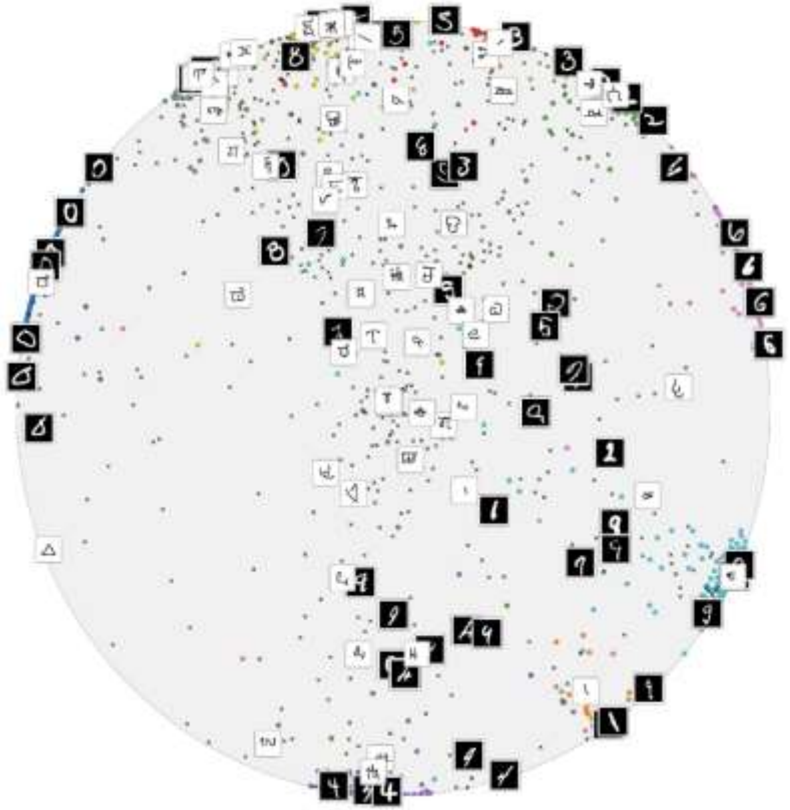
Encoder	Dataset			
	CIFAR10	CIFAR100	CUB	*Mini*ImageNet
Inception v3 [49]	0.25	0.23	0.23	0.21
ResNet34 [14]	0.26	0.25	0.25	0.21
VGG19 [42]	0.23	0.22	0.23	0.17

Images are naturally hierarchical, hyperbolic embeddings improve few-shot learning.

Hyperbolic Image Embeddings

[Khrulkov et al. CVPR 2020]



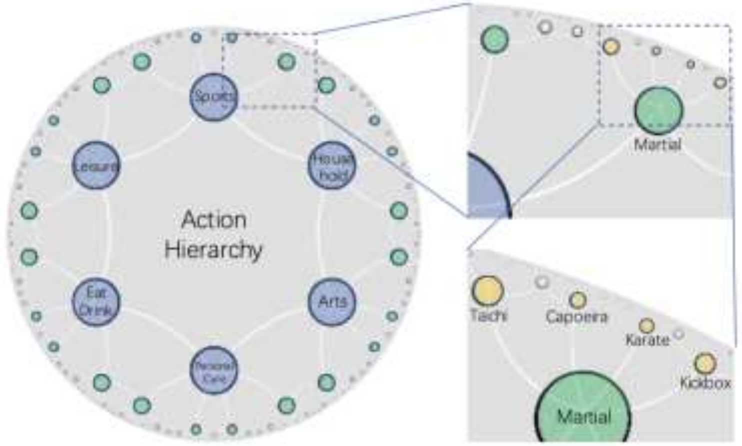


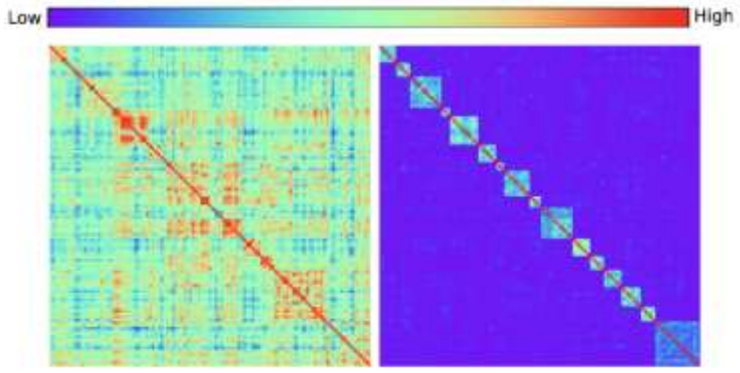
Encoder	Dataset			
	CIFAR10	CIFAR100	CUB	*Mini*ImageNet
Inception v3 [49]	0.25	0.23	0.23	0.21
ResNet34 [14]	0.26	0.25	0.25	0.21
VGG19 [42]	0.23	0.22	0.23	0.17

Images are naturally hierarchical, hyperbolic embeddings improve few-shot learning.

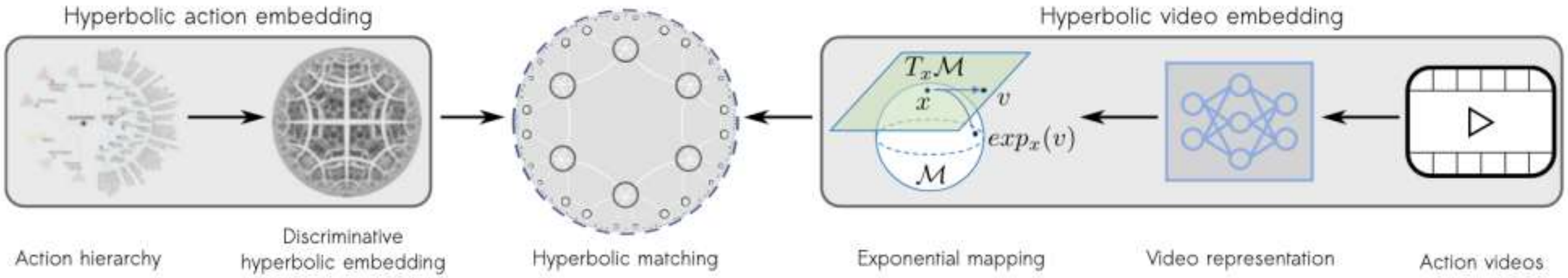
Hyperbolic actions

[Long et al.. CVPR 2020]





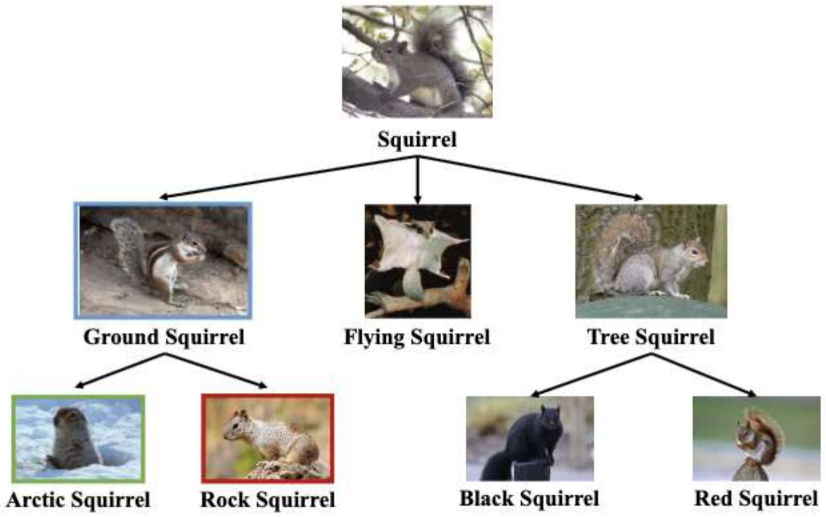
(a) One-hot. (b) Hyperbolic (*ours*).

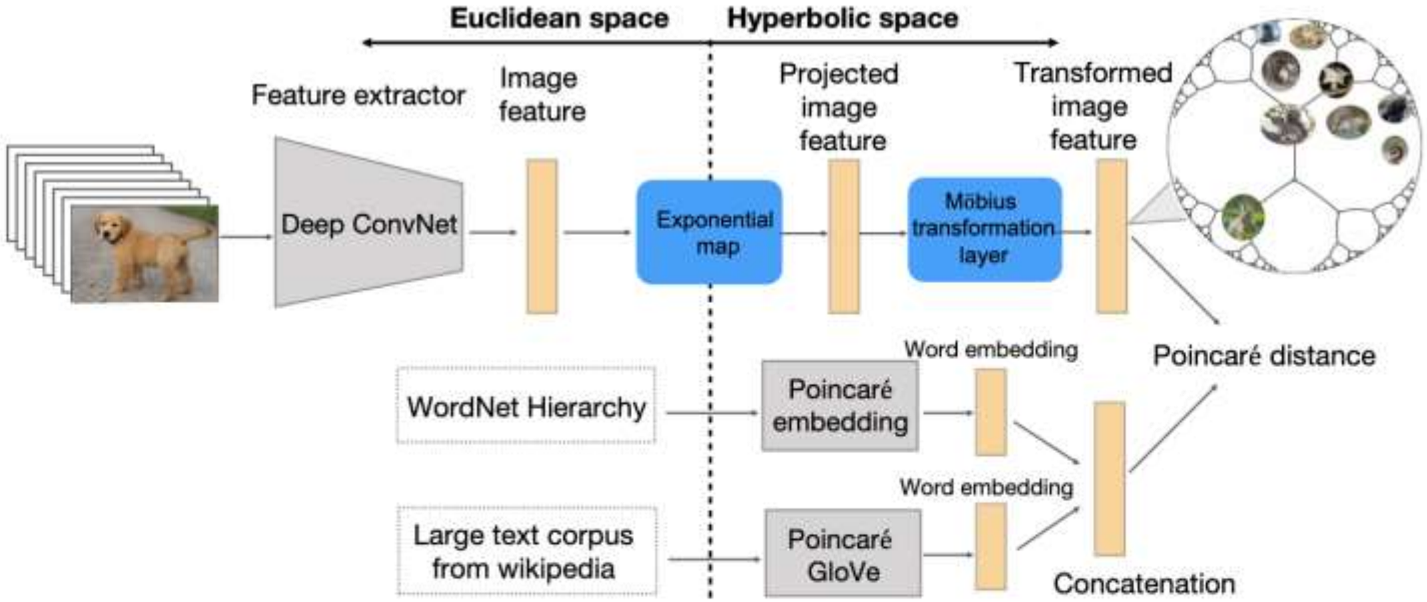


Videos are naturally hierarchical, hyperbolic embeddings improve action recognition.

Hyperbolic zero-shot learning

[Liu et al. CVPR 2020]

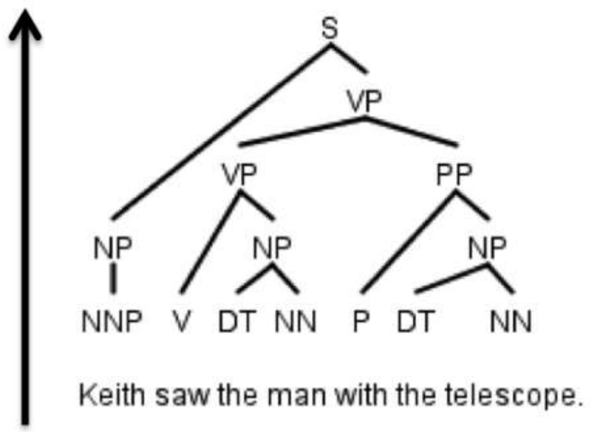


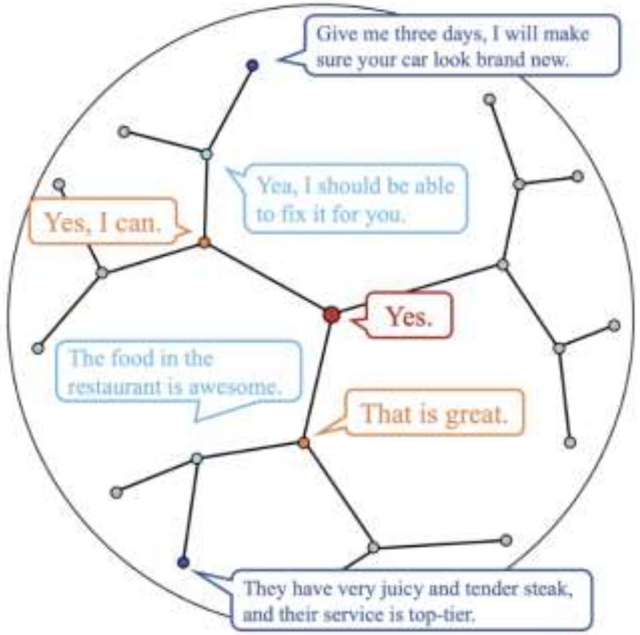


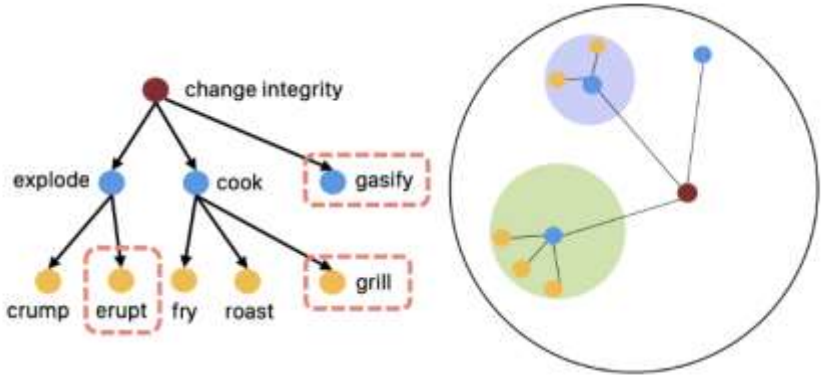
Semantics is naturally hierarchical, hyperbolic embeddings improve zero-shot recognition.

Hyperbolic embeddings for text

Text is well-known to be hierarchical at multiple levels.



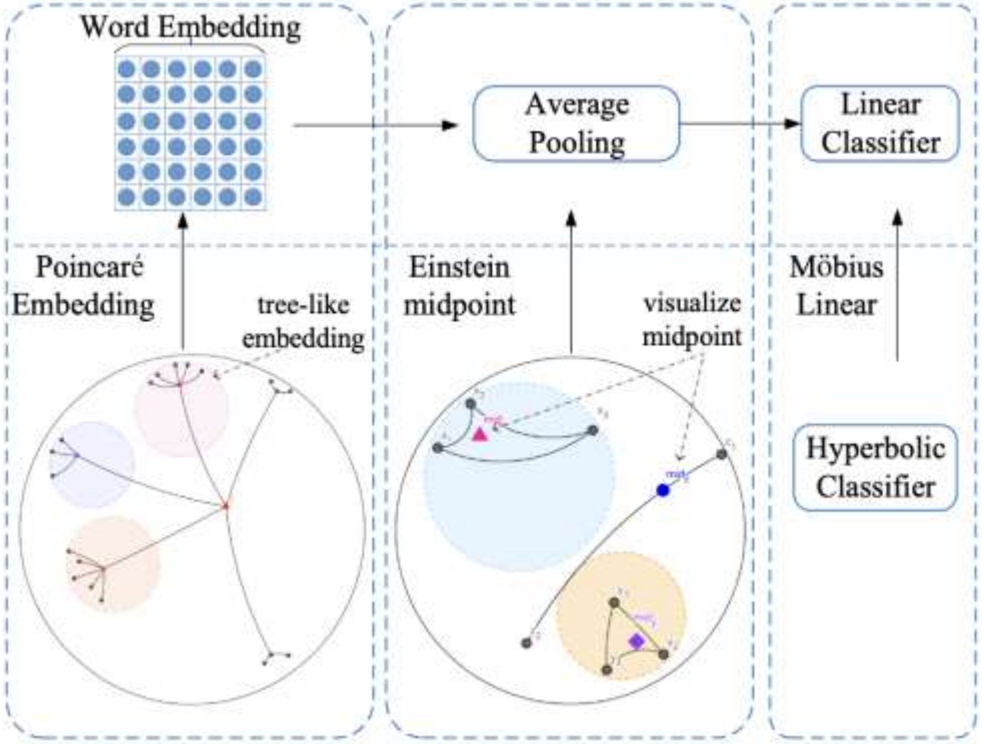




Should representations of text then also be embedded in a hierarchical geometry?

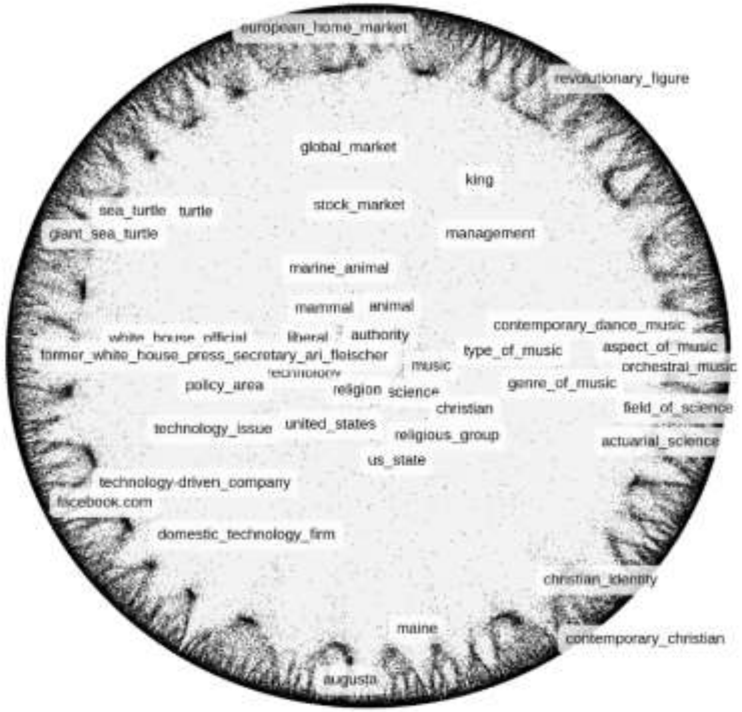
Hyperbolic word embeddings

[Zhu et al. 2021]



Hyperbolic FastText

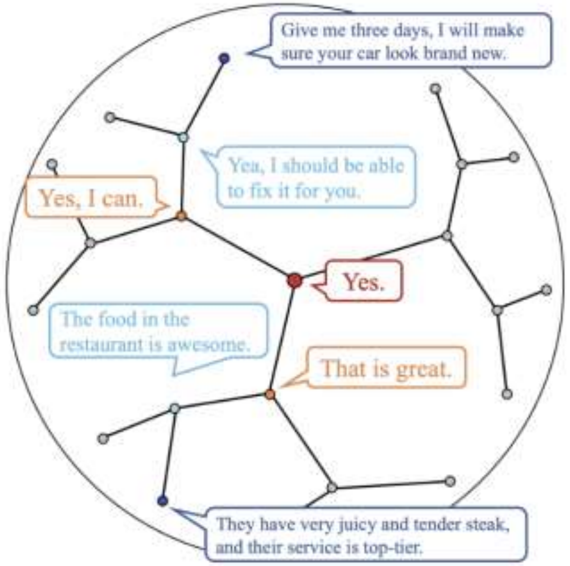
[Le et al. 2019]

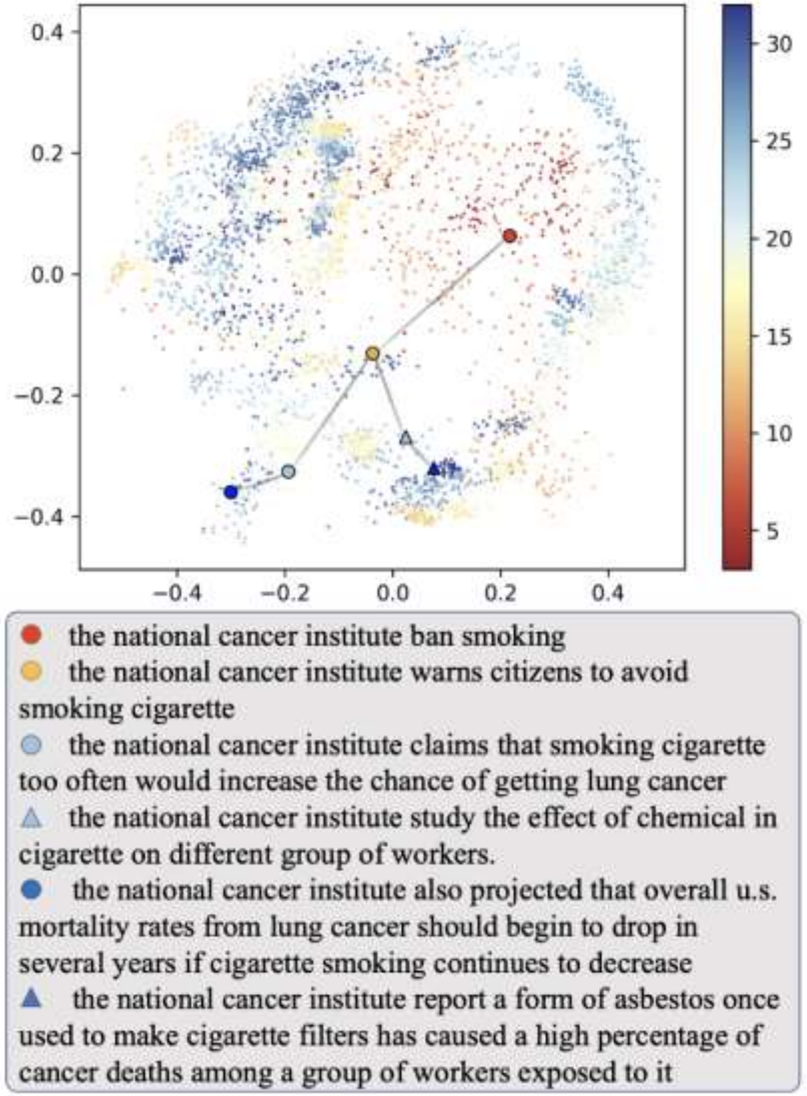


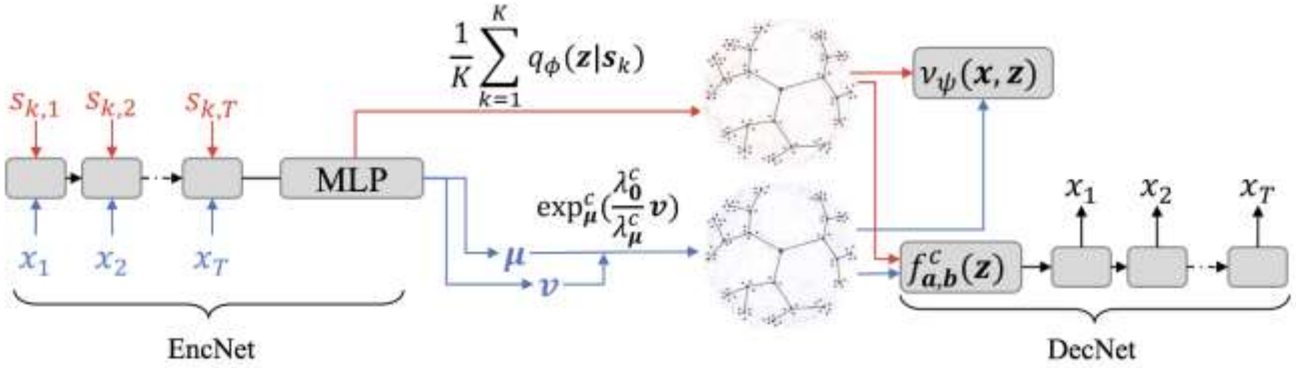
Inferring concept hierarchies from text in hyperbolic space

Hyperbolic generation of text

[Dai et al. 2020]







Hyperbolic vision-language models

[Desai et al. ICML 2023]

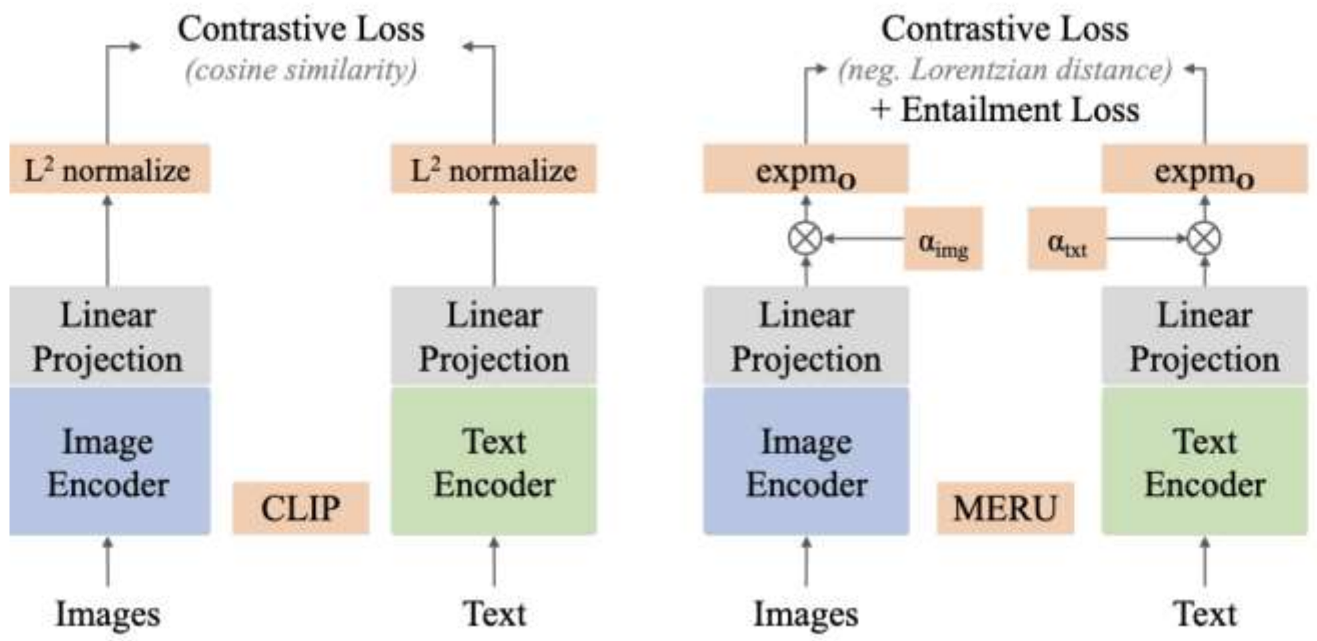


Image-text representation learning wants to collapse image and text embeddings.

Issue 1: image-text asymmetry

Vision-language models explicitly assume that image and text are equal.

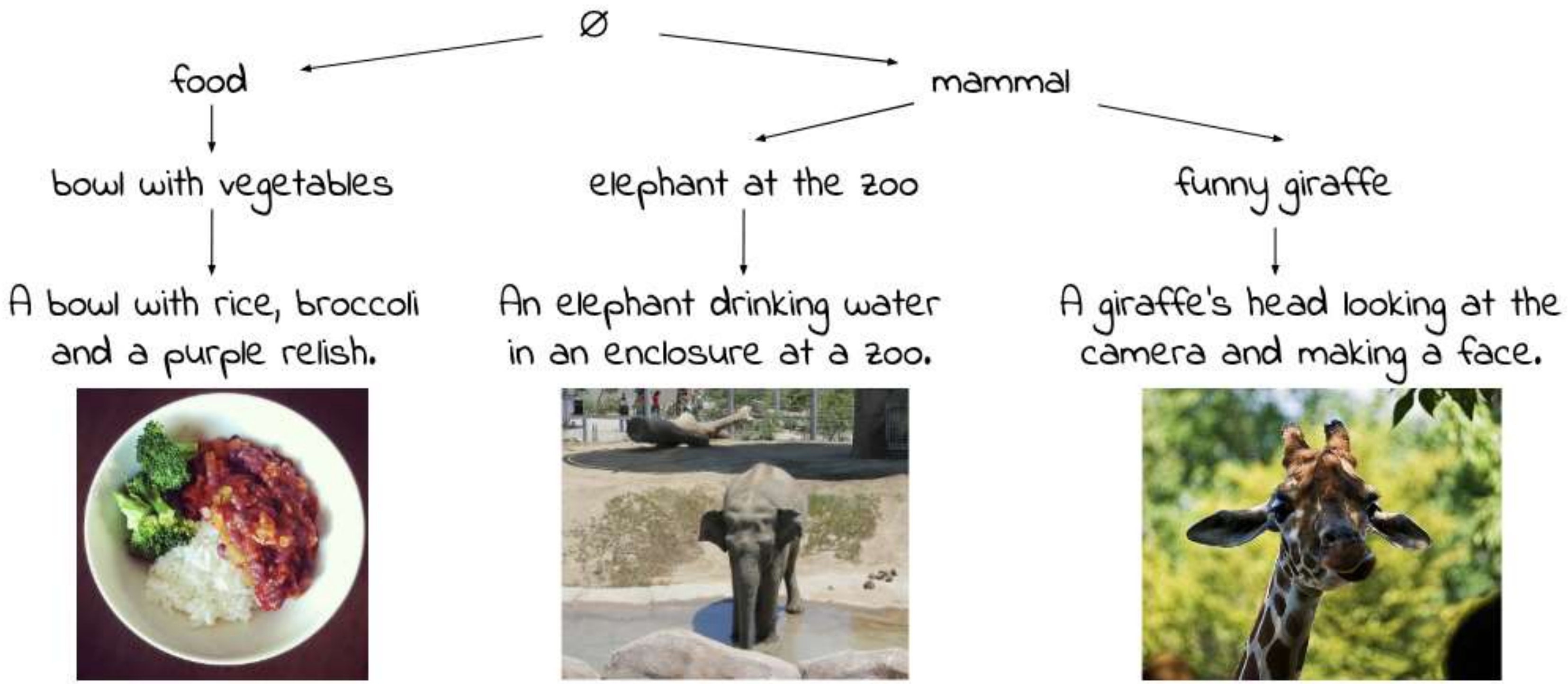
Both are points in a shared space, contrastive learning wants to collapse them.

But one sentence does not unique define an image.

Text is more general than images.

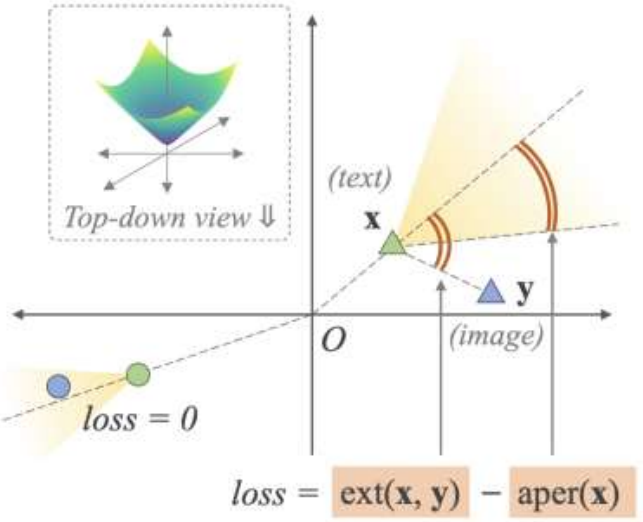
Issue 2: vision-language is hierarchical

"Foundation VLMs exhibit zero-shot hierarchical understanding" – Alper et al. (2024)



Hyperbolic vision-language models

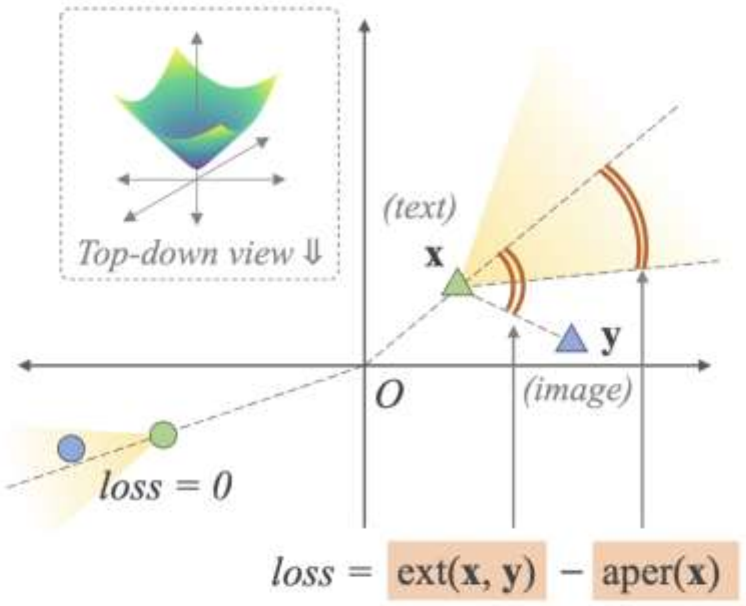
Intuitively, image and text embeddings are unequal!



Hyperbolic entailments allow to model this imbalance
and learn the hierarchical nature of image-text representations.

Entailment as a loss

Main objective: model the asymmetry between image and text.



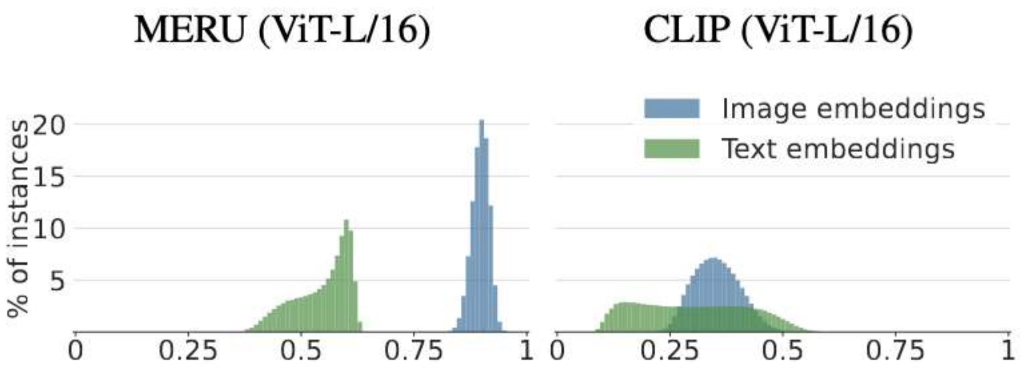
Hyperbolic entailments allow to model this imbalance

and (implicitly) learn the hierarchical nature of image-text representations.

Empirical effect

		text → image				*image → text*			
		COCO		Flickr		COCO		Flickr	
		R5	R10	R5	R10	R5	R10	R5	R10
ViT S/16	CLIP	29.9	40.1	35.3	46.1	37.5	48.1	42.1	54.7
	MERU	**30.5**	**40.9**	**37.1**	**47.4**	**39.0**	**50.5**	**43.5**	**55.2**
ViT B/16	CLIP	32.9	43.3	40.3	51.0	41.4	52.7	**50.2**	**60.2**
	MERU	**33.2**	**44.0**	**41.1**	**51.6**	**41.8**	**52.9**	48.1	58.9
ViT L/16	CLIP	31.7	42.2	39.0	49.3	40.6	51.3	47.8	58.5
	MERU	**32.6**	**43.0**	**39.6**	**50.3**	**41.9**	**53.3**	**50.3**	**60.6**

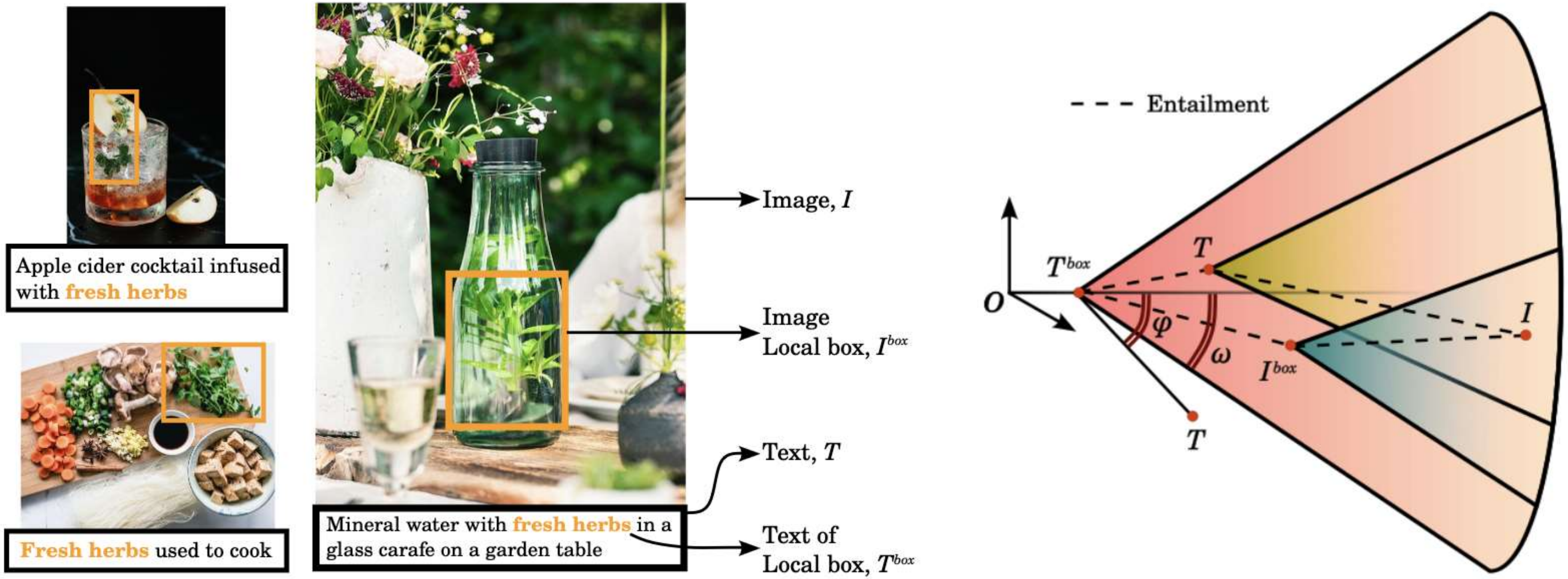
		Embedding width				
		512	256	128	96	64
COCO *text→image*	CLIP	31.7	31.8	31.4	29.6	25.7
	MERU	**32.6**	**32.7**	**32.7**	**31.0**	**26.5**
COCO *image→text*	CLIP	40.6	41.0	40.4	37.9	33.3
	MERU	**41.9**	**42.5**	**42.6**	**40.5**	**34.2**
ImageNet	CLIP	38.4	38.3	37.9	35.2	30.2
	MERU	**38.8**	**38.8**	**38.8**	**37.3**	**32.3**



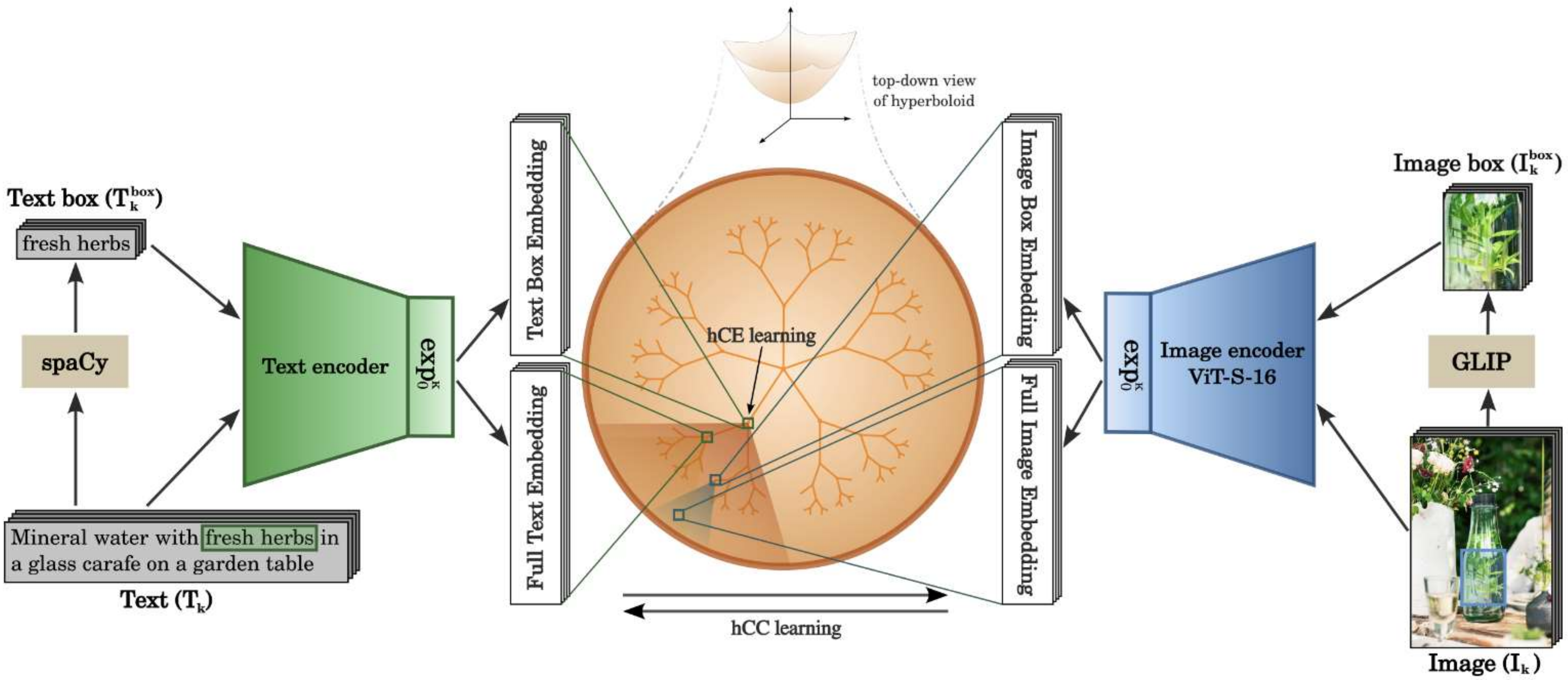
$d(\mathbf{z}) = \|\mathbf{z}_{space}\|$ $\quad$ $d(\mathbf{z}) = 0.5(1 - \langle \mathbf{z}, \texttt{[ROOT]} \rangle)$

Compositional vision-language models

[Pal et al. ICLR 2025 (oral)]

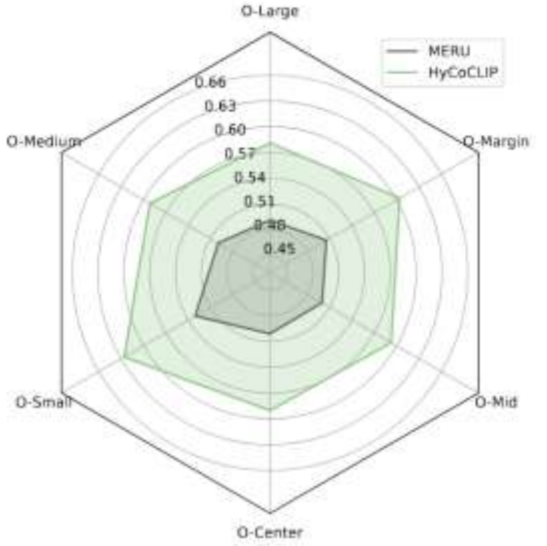


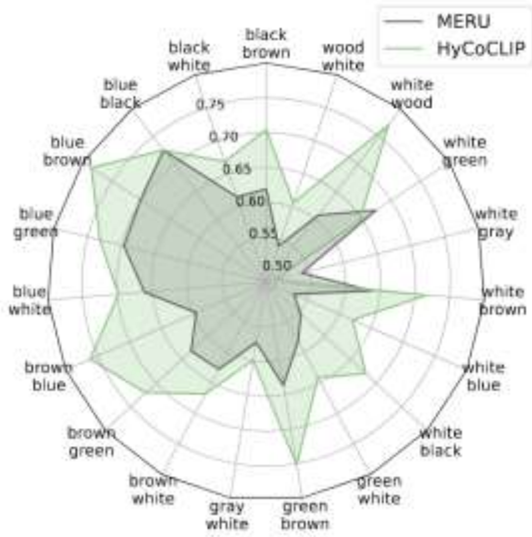
Multi-modal network



Entailment compositions work

		w/ boxes	samples (M)	General datasets						Fine-grained datasets						Misc. datasets			
				ImageNet	CIFAR-10	CIFAR-100	SUN397	Caltech-101	STL-10	Food-101	CUB	Cars	Aircraft	Pets	Flowers	DTD	EuroSAT	RESISC45	Country211
	RedCaps																		
ViT S/16	CLIP[†]	✗	11.4	32.5	66.7	35.8	26.7	60.8	89.8	72.5	29.8	11.1	1.3	72.5	44.9	16.4	30.1	27.7	5.0
	CLIP	✓	11.4 [6.3]	30.2	76.5	42.4	25.8	62.3	89.5	69.6	25.7	8.5	2.2	65.3	38.6	13.6	36.6	28.5	4.6
	MERU[†]	✗	11.4	31.4	65.9	35.2	26.8	58.1	89.3	71.4	29.0	8.3	1.6	71.0	40.9	17.0	29.9	29.3	4.7
	MERU	✓	11.4 [6.3]	29.9	76.4	39.9	26.6	62.3	89.5	68.4	25.4	8.9	1.2	67.2	37.6	13.0	30.5	27.6	4.3
	HyCoCLIP	✓	5.8 [6.3]	31.9	77.4	37.7	27.6	64.5	90.9	71.1	28.8	9.7	1.1	70.5	41.4	13.4	22.7	30.7	4.4
	GRIT																		
ViT S/16	CLIP	✗	20.5	36.7	70.2	42.6	49.5	73.6	89.7	44.7	9.8	6.9	2.0	44.6	14.8	22.3	**40.7**	40.1	5.1
	CLIP	✓	20.5 [35.9]	36.2	84.2	54.8	46.1	74.1	91.6	43.2	11.9	6.0	2.5	45.9	18.1	24.0	32.4	35.5	4.7
	MERU	✗	20.5	35.4	71.2	42.0	48.6	73.0	89.8	48.8	10.9	6.5	2.3	42.7	17.3	18.6	39.1	38.9	5.3
	MERU	✓	20.5 [35.9]	35.0	85.0	54.0	44.6	73.9	91.6	41.1	10.1	5.6	2.2	43.9	15.9	24.5	39.3	33.5	4.8
	HyCoCLIP	✓	20.5 [35.9]	41.7	85.0	53.6	52.5	75.7	92.5	50.2	14.7	8.1	**4.2**	52.0	20.5	22.3	33.8	**45.7**	5.2
ViT B/16	CLIP	✗	20.5	40.6	78.9	48.3	53.0	76.7	92.4	48.6	10.0	9.0	3.4	45.9	21.3	23.4	37.1	42.7	5.7
	MERU	✗	20.5	40.1	78.6	49.3	53.0	72.8	93.2	51.5	11.9	8.6	3.7	48.5	21.2	22.2	31.7	44.2	5.6
	HyCoCLIP	✓	20.5 [35.9]	**45.8**	**88.8**	**60.1**	**57.2**	**81.3**	**95.0**	**59.2**	**16.4**	**11.6**	3.7	**56.8**	**23.9**	**29.4**	35.8	45.6	**6.5**

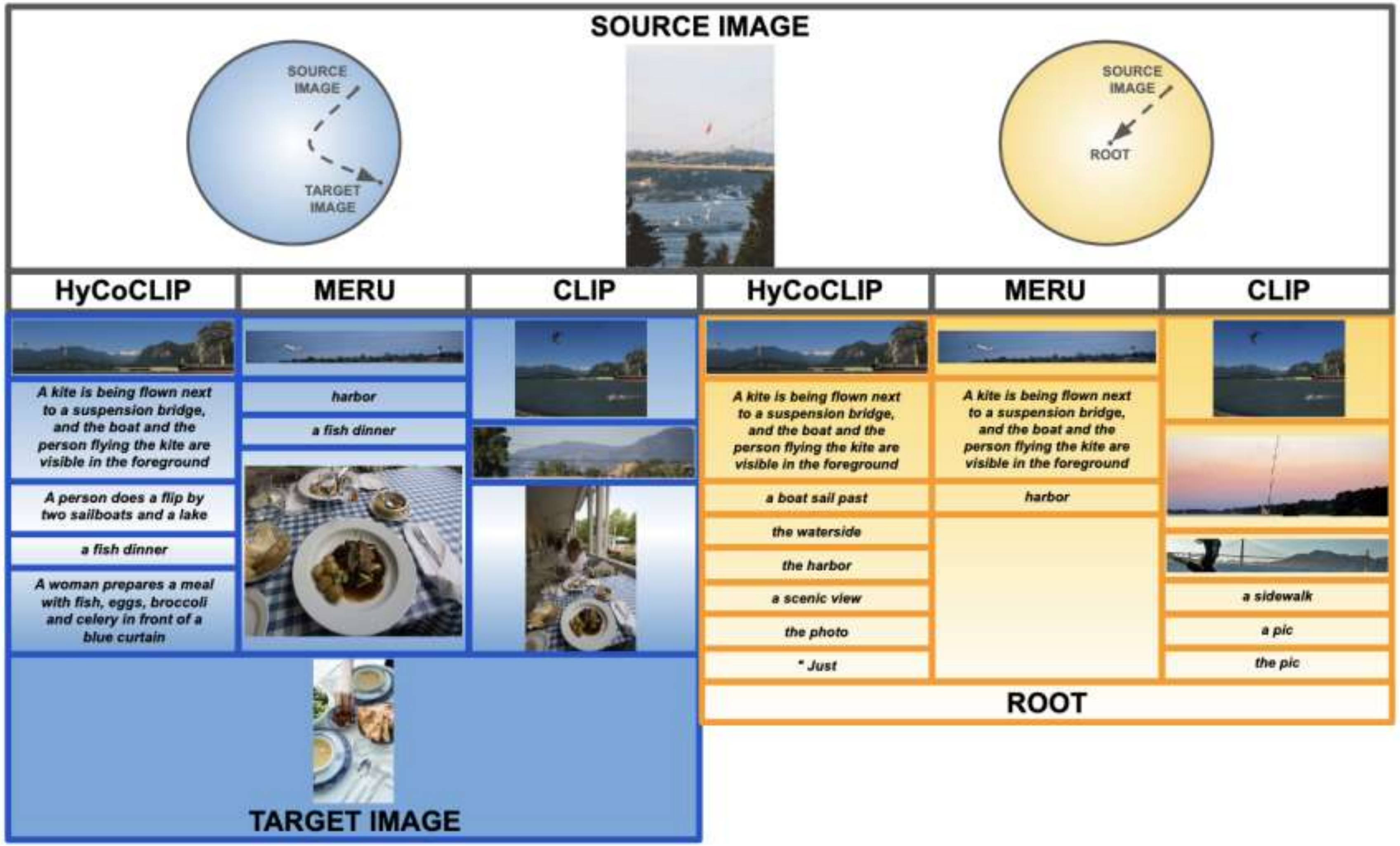


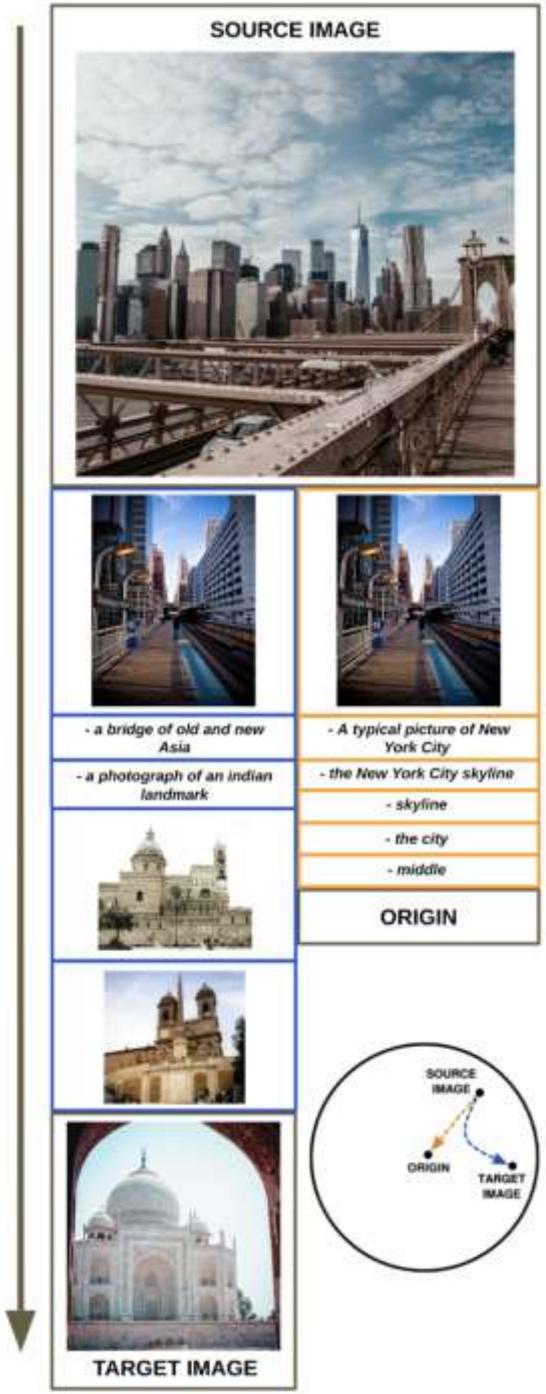


Vision encoder	Model	w/ boxes	Text retrieval				Image retrieval				Hierarchical metrics				
			COCO		Flickr		COCO		Flickr		WordNet				
			R@5	R@10	R@5	R@10	R@5	R@10	R@5	R@10	TIE(↓)	LCA(↓)	J(↑)	P_H(↑)	R_H(↑)
ViT S/16	CLIP	✗	69.3	79.1	**90.2**	**95.2**	53.7	65.2	81.1	87.9	4.02	2.39	0.76	0.83	0.84
	CLIP	✓	60.7	71.8	84.2	91.3	47.1	58.6	73.1	82.1	4.03	2.38	0.76	0.83	0.83
	MERU	✗	68.8	78.8	89.4	94.8	53.6	65.3	80.4	87.5	4.08	2.39	0.76	0.83	0.83
	MERU	✓	**72.7**	**81.9**	83.5	90.1	46.6	58.3	60.0	71.7	4.08	2.39	0.75	0.83	0.83
	HyCoCLIP	✓	69.5	79.5	89.1	93.9	**55.2**	**66.6**	**81.5**	**88.1**	**3.55**	**2.17**	**0.79**	**0.86**	**0.85**
ViT B/16	CLIP	✗	71.4	81.5	**93.6**	**96.9**	57.4	68.5	83.5	89.9	3.60	2.21	0.79	0.85	0.85
	MERU	✗	**72.3**	82.0	93.5	96.2	57.4	68.6	84.0	90.0	3.63	2.22	0.78	0.85	0.85
	HyCoCLIP	✓	72.0	**82.0**	92.6	95.4	**58.4**	**69.3**	**84.9**	**90.3**	**3.17**	**2.05**	**0.81**	**0.87**	**0.87**

Model	AP
CLIP	51.2
MERU	55.8
RegionCLIP	65.2
HyCoCLIP	**68.5**

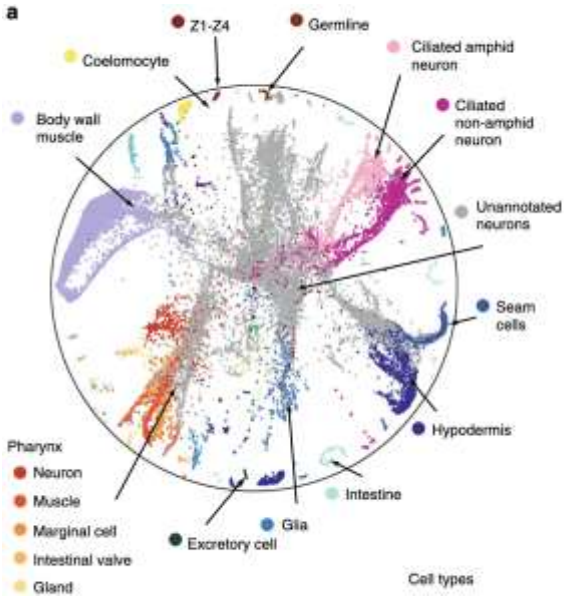
Embedding traversal





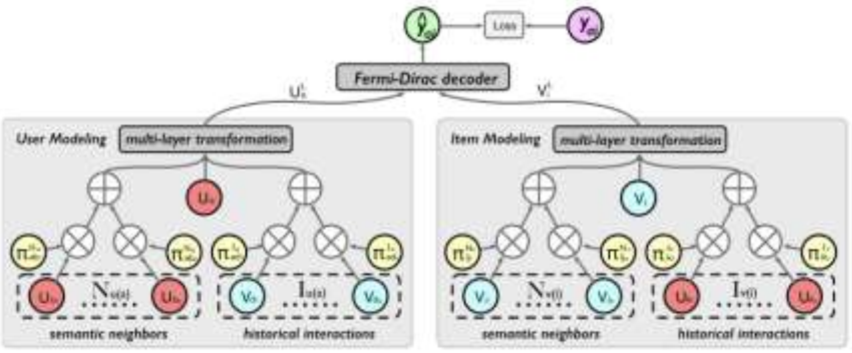
Hyperbolic embeddings for other data types

[Klimovskaia et al. Nature Comm. 2020]



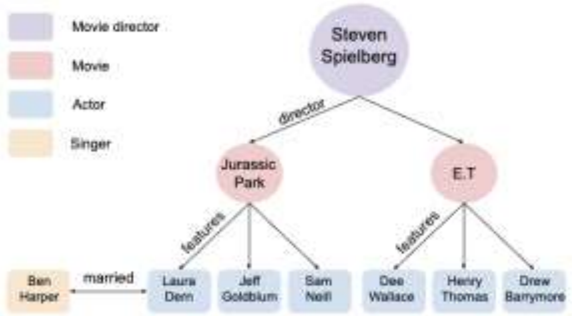
Hyperbolic embeddings of single-cell data.

[Li et al. TKDD 2023]



Hyperbolic recommender systems

[Chami et al. 2020]



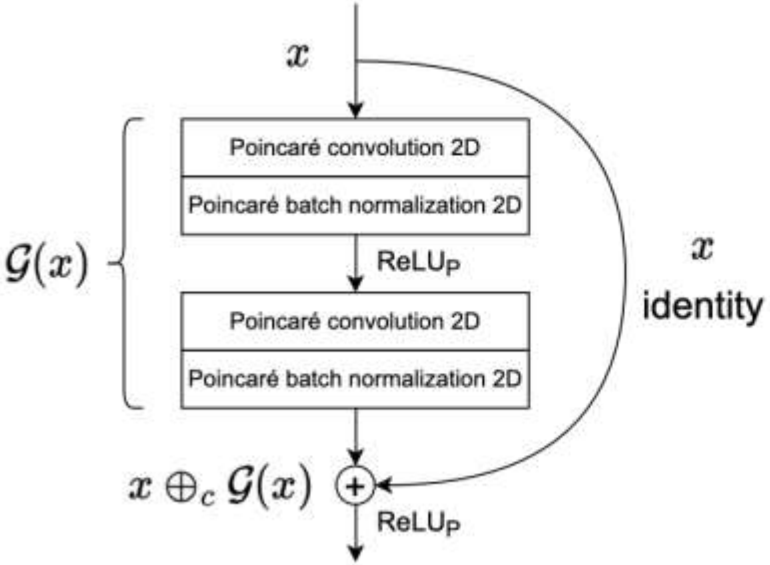
Hyperbolic knowledge graphs

And more for music, 3D skeletons, phylogenetic placement, social networks, clustering…

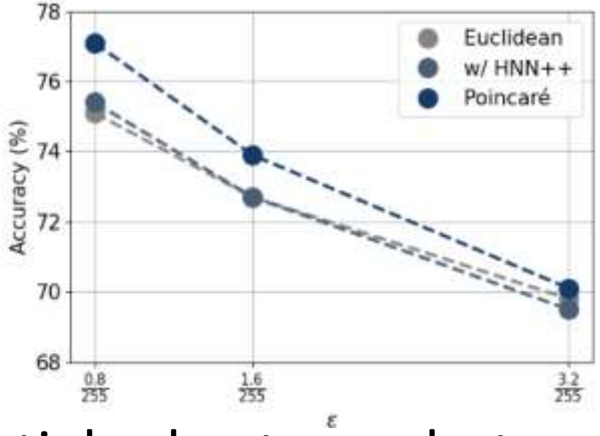
What if every layer becomes hyperbolic?

[van Spengler et al. ICCV 2023]

First convolutional network for images fully in hyperbolic space.



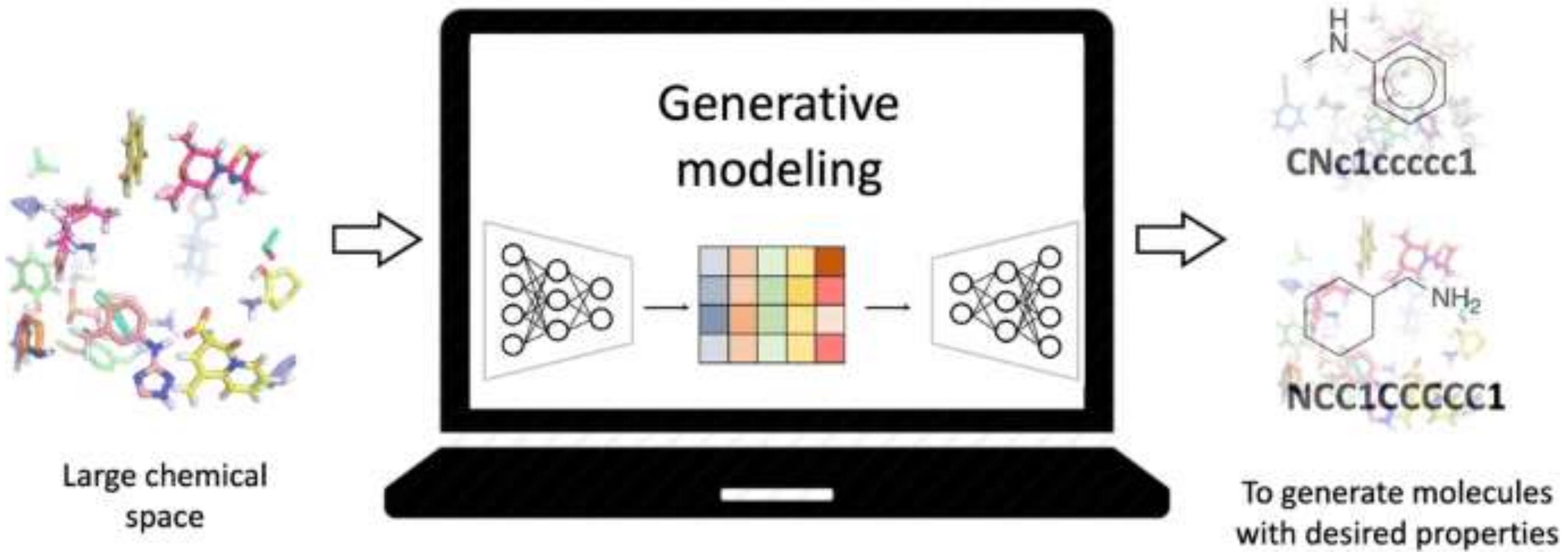
	Manifold	CIFAR-10						CIFAR-100					
		FPR95 ↓		AUROC ↑		AUPR ↑		FPR95 ↓		AUROC ↑		AUPR ↑	
		R20	R32	R20	R32	R20	R32	R20	R32	R20	R32	R20	R32
Places-365	Euclidean	64.2	72.3	**84.7**	82.0	**96.2**	95.6	89.5	93.9	62.5	57.9	89.3	87.9
	w/ HNN++	**63.8**	72.7	79.6	77.7	94.5	94.2	93.2	86.3	63.3	66.6	89.8	91.1
	Poincaré	70.2	**70.7**	82.3	**82.6**	95.7	**95.9**	**82.8**	**83.8**	**71.5**	**71.1**	**92.3**	**92.2**
				68.8	73.4	92.8	94.1	99.5	98.8	43.7	54.6	83.7	88.2
				85.5	82.2	**96.9**	96.1	92.1	88.6	66.4	68.9	91.1	92.0
				85.0	**83.6**	96.6	**96.3**	**76.9**	**83.0**	**76.8**	**72.6**	**94.1**	**92.9**
				73.6	77.3	93.2	94.7	98.1	96.0	33.5	42.9	75.9	79.4
				79.6	**85.8**	94.5	**96.6**	85.9	**77.5**	58.9	65.7	86.8	89.0
				82.1	82.3	**95.5**	95.6	**83.9**	84.2	**67.7**	**68.8**	**91.0**	**91.5**



It works! Also better OOD and adversatial robustness, but scaling remain challenging.

Hyperbolic generative learning

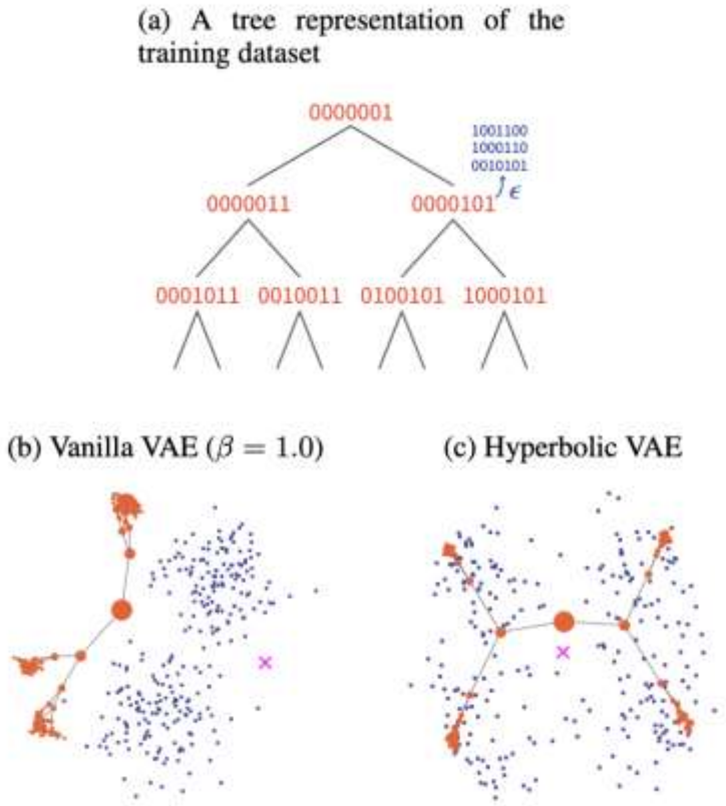
[Bian and Xie, JMM 2021]

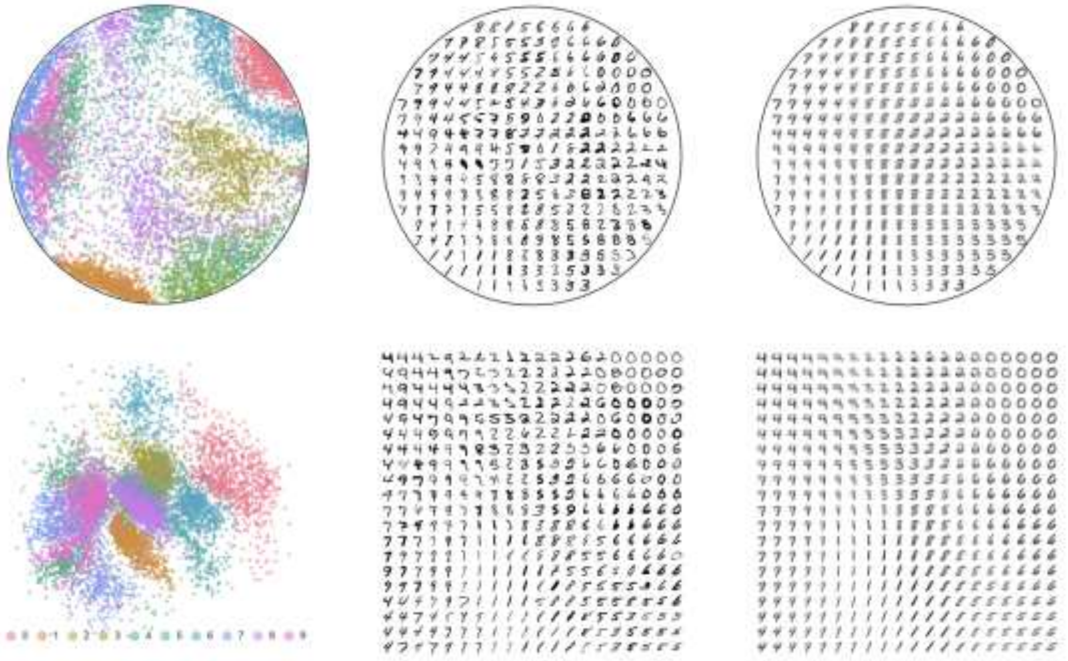


So far, we focused on discriminative learning. What about generation in hyperbolic space?

Hyperbolic Variational Autoencoders

[Nagano et al. ICML 2019, Mathieu et al. NeurIPS 2019]



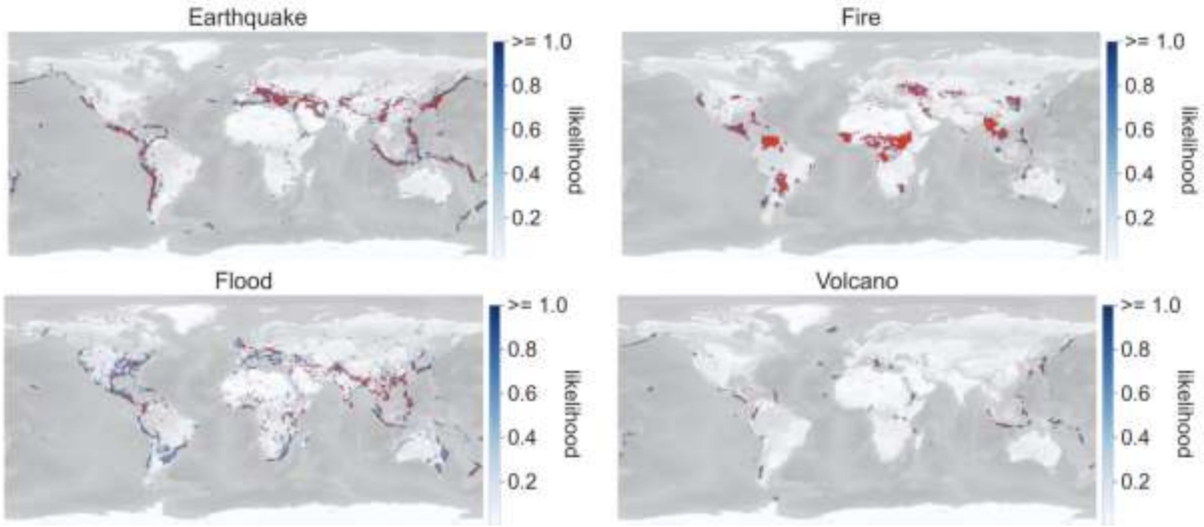


Hyperbolic latent spaces with wrapped normals enable learning latent hierarchical distributions.

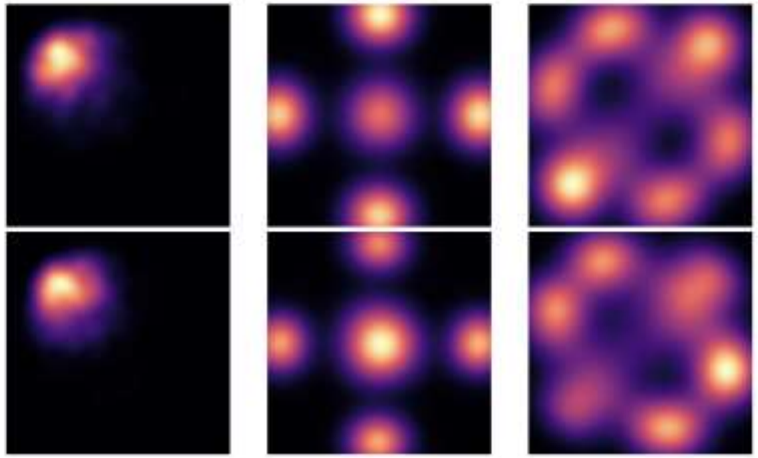
Riemannian diffusion models

[Huang et al. NeurIPS 2022]

Generalize diffusion process to any Riemannian manifold.



Spherical



Hyperbolic

Allows us to adapt diffusion process to the real underlying data distribution.

The big potential of hyperbolic learning

Hierarchical learning model the hierarchies of semantics and data.

Robust learning handle new distributions and adversarial samples.

Low-dimensional learning hyperbolic space is dense, allowing for smaller networks.

Brain-like networks brains are likely hyperbolic, big links with neuroscience

Zhang et al. Nature Communications 2022

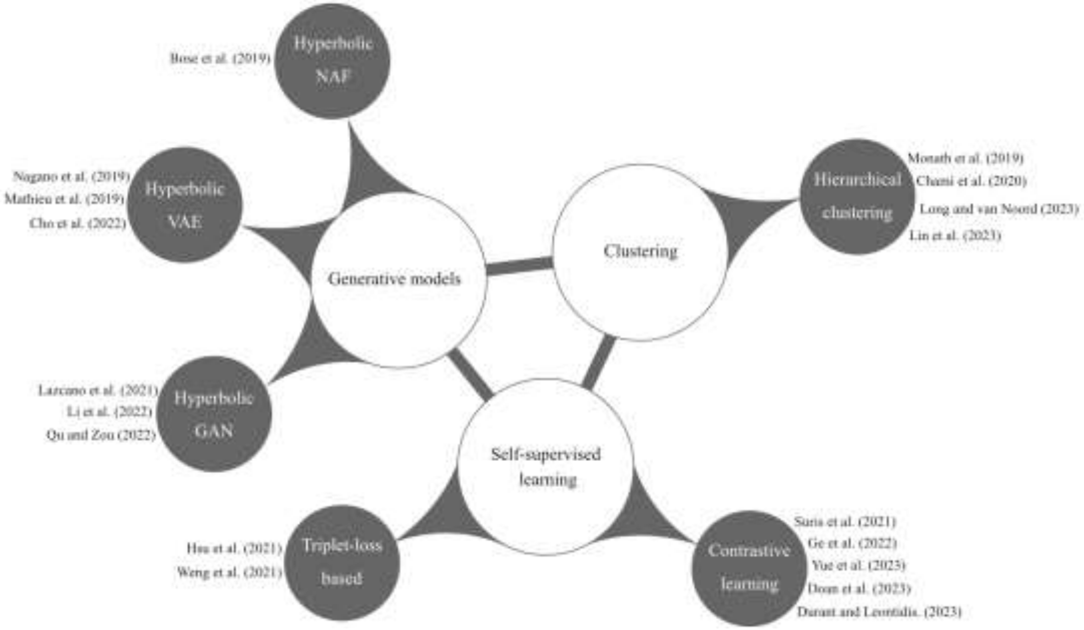
The grand challenges of hyperbolic learning

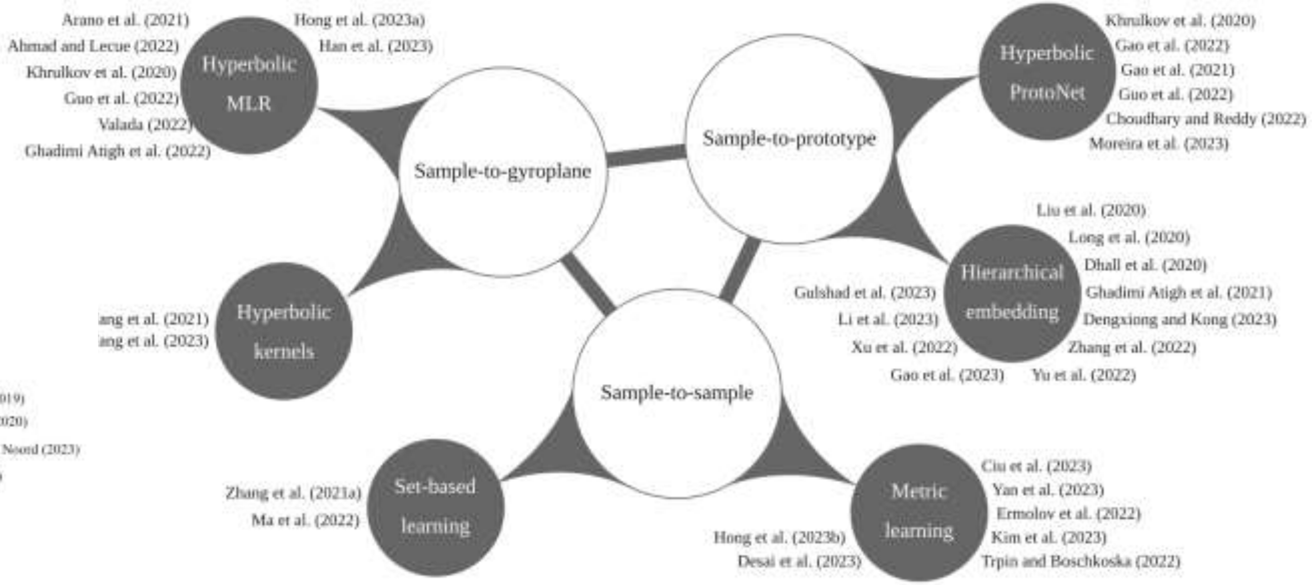
Fully hyperbolic learning	which hyperbolic model is best? and how to optimize?
Computational challenges	numerical stability and speed of computation.
Open source community	where is hyperbolic PyTorch?
Learning at scale	we need an ImageNet/CLIP moment for hyperbolic learning.

Where to start?

Pascal Mettes, Mina Ghadimi, Martin Keller-Ressel, Jeffrey Gu, Serena Yeung. IJCV 2024

Hyperbolic Deep Learning in Computer Vision: A Survey





Where to start?

Max van Spengler, Philipp Wirth, Pascal Mettes. ACM MM 2024

HypLL: The Hyperbolic Learning Library

```python
import hypll.nn as hnn
from hypll.manifolds.poincare_ball import (
    Curvature, PoincareBall
)

ball = PoincareBall(Curvature(1.0))
class HNet(nn.Module):
    def __init__(self):
        super().__init__()
        self.conv1 = hnn.HConvolution2d(3, 6, 5, ball)
        self.pool = hnn.HMaxPool2d(2, ball, 2)
        self.conv2 = hnn.HConvolution2d(6, 16, 5, ball)
        self.fc1 = hnn.HLinear(16 * 5 * 5, 120, ball)
        self.fc2 = hnn.HLinear(120, 84, ball)
        self.fc3 = hnn.HLinear(84, 10, ball)
        self.relu = hnn.HReLU(ball)

    def forward(self, x):
        x = self.pool(self.relu(self.conv1(x)))
        x = self.pool(self.relu(self.conv2(x)))
        x = x.flatten(1)
        x = self.relu(self.fc1(x))
        x = self.relu(self.fc2(x))
        x = self.fc3(x)
        return x
```

```python
import torch.nn as nn



class Net(nn.Module):
    def __init__(self):
        super().__init__()
        self.conv1 = nn.Conv2d(3, 6, 5)
        self.pool = nn.MaxPool2d(2, 2)
        self.conv2 = nn.Conv2d(6, 16, 5)
        self.fc1 = nn.Linear(16 * 5 * 5, 120)
        self.fc2 = nn.Linear(120, 84)
        self.fc3 = nn.Linear(84, 10)
        self.relu = nn.ReLU()

    def forward(self, x):
        x = self.pool(self.relu(self.conv1(x)))
        x = self.pool(self.relu(self.conv2(x)))
        x = x.flatten(1)
        x = self.relu(self.fc1(x))
        x = self.relu(self.fc2(x))
        x = self.fc3(x)
        return x
```

https://github.com/maxvanspengler/hyperbolic_learning_library

Where to start?

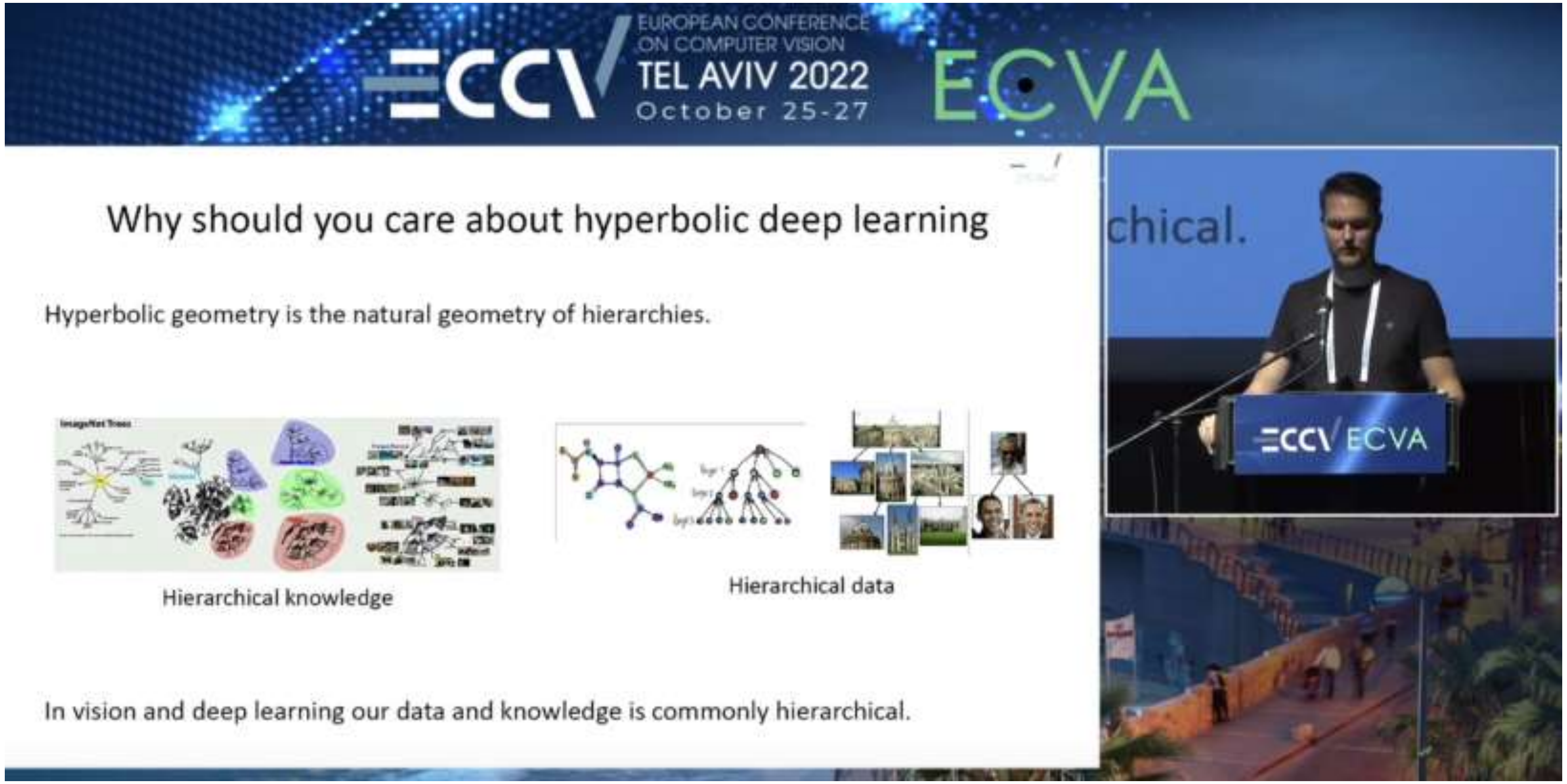
Max van Spengler, Philipp Wirth, Pascal Mettes. ACM MM 2024

HypLL: The Hyperbolic Learning Library

```
from hypll.tensors import TangentTensor

for data in trainloader:
    inputs, labels = data

    tangents = TangentTensor(
        inputs, man_dim=1, manifold=ball
    )
    inputs_on_ball = ball.expmap(tangents)

    outputs = hnet(inputs_on_ball)
```

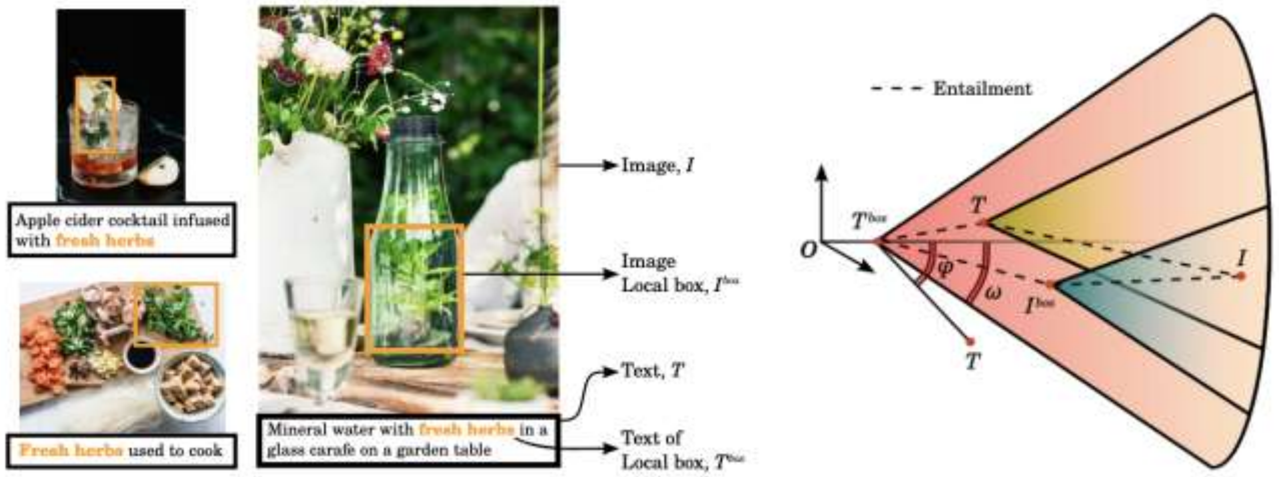
https://github.com/maxvanspengler/hyperbolic_learning_library

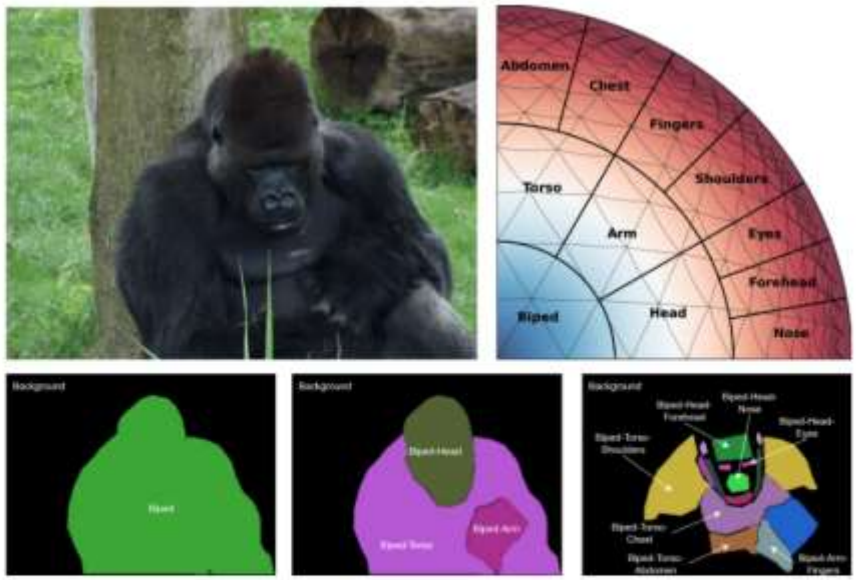
Where to start?

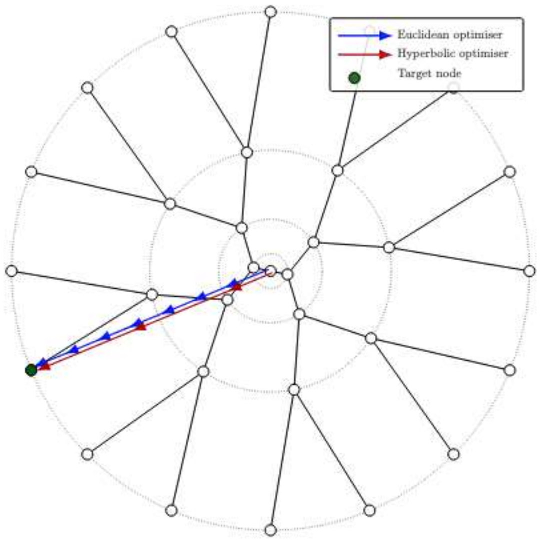


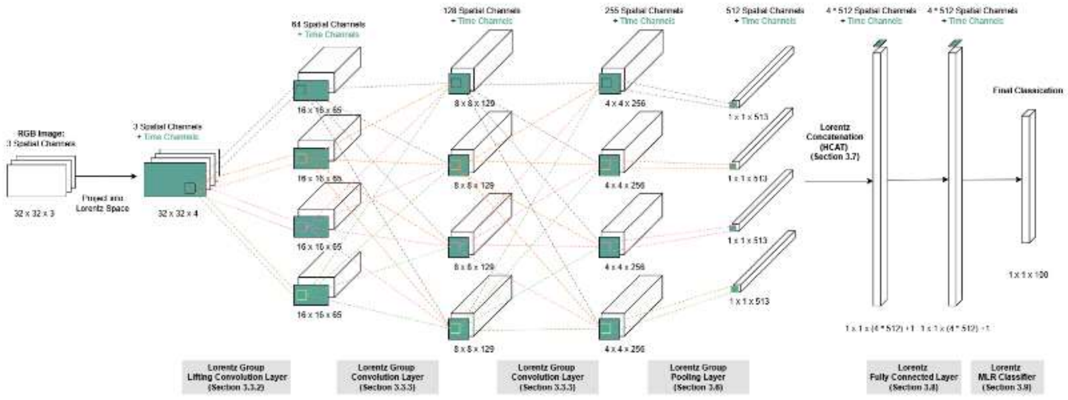
https://www.youtube.com/@hyperboliclearningforcv/playlists

Interested? Let’s write a hyperbolic thesis!









Next lecture

Lecture	Title	Lecture	Title
1	Intro and history of deep learning	2	AutoDiff
3	Deep learning optimization I	4	Deep learning optimization II
5	Convolutional deep learning	6	Attention-based deep learning
7	Graph deep learning	8	From supervised to unsupervised deep learning
9	Multi-modal deep learning	10	Generative deep learning
11	What doesn't work in deep learning	12	Non-Euclidean deep learning
13	Q&A	14	Deep learning for videos