Hyperbolic Image-Text Representations
(ICML 2023)

Karan Desai, Maximilian Nickel, Tanmay Rajpurohit, Justin Johnson, Ramakrishna Vedantam

October 1, 2025

Seoul National University - IDEA Lab.
Presenter: Sehyun Park

Introduction

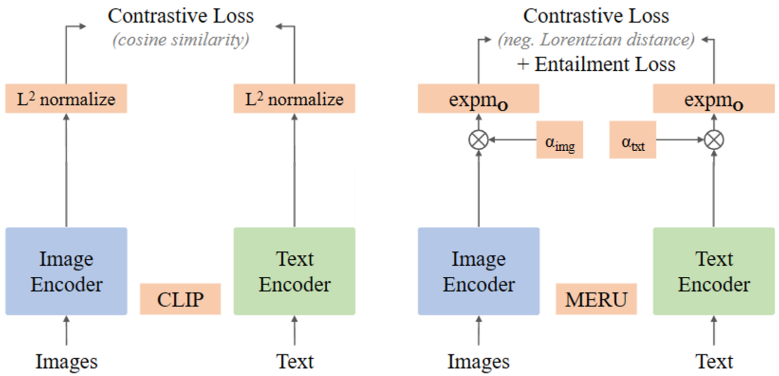
- Visual and linguistic concepts naturally organize themselves in a hierarchy, where a textual concept “dog” entails all images that contain dogs.

- Current large-scale vision and language models such as CLIP do not capture this hierarchy.

▶ In this paper, the authors introduce **MERU**, the first large-scale image-text models that leverage a hyperbolic embedding space.

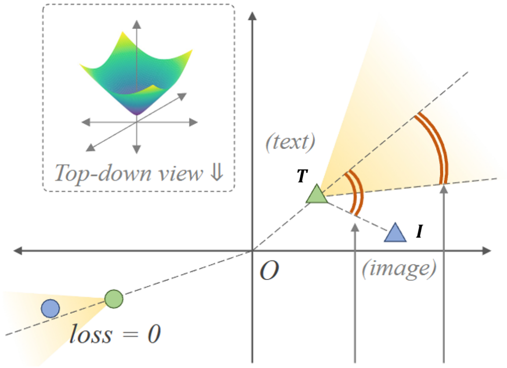
Methods

- CLIP vs. MERU: Training Comparison



Methods

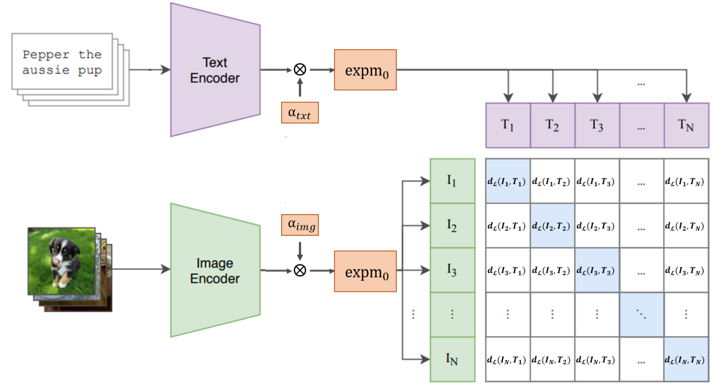
- Entailment Loss



$$\textit{Entailment loss} = \max(0, \boxed{\mathrm{ext}(T, I)} - \boxed{\mathrm{aper}(T)})$$

Methods

▶ Contrastive Loss



Pepper the aussie pup
Text Encoder
α_{txt}
expm_0
Image Encoder
α_{img}
expm_0
T_1 T_2 T_3 ... T_N
I_1 I_2 I_3 ⋮ I_N
$d_{\mathcal{L}}(I_1,T_1)$ $d_{\mathcal{L}}(I_1,T_2)$ $d_{\mathcal{L}}(I_1,T_3)$ ... $d_{\mathcal{L}}(I_1,T_N)$
$d_{\mathcal{L}}(I_2,T_1)$ $d_{\mathcal{L}}(I_2,T_2)$ $d_{\mathcal{L}}(I_2,T_3)$ ... $d_{\mathcal{L}}(I_2,T_N)$
$d_{\mathcal{L}}(I_3,T_1)$ $d_{\mathcal{L}}(I_3,T_2)$ $d_{\mathcal{L}}(I_3,T_3)$ ... $d_{\mathcal{L}}(I_3,T_N)$
$d_{\mathcal{L}}(I_N,T_1)$ $d_{\mathcal{L}}(I_N,T_2)$ $d_{\mathcal{L}}(I_N,T_3)$ ... $d_{\mathcal{L}}(I_N,T_N)$

$$\textit{Contrastive loss} = \sum_{i=j} d_{\mathcal{L}}(I_i, T_j) - \sum_{i\neq j} d_{\mathcal{L}}(I_i, T_j)$$

※ $d_{\mathcal{L}}(\cdot,\cdot)$ is *Lorentzian distance.*

Lorentzian model

- Every vector $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^{d+1}$ can be written as $[\boldsymbol{x}_{space}, x_{time}]$, $[\boldsymbol{y}_{space}, y_{time}]$, where $\boldsymbol{x}_{space}, \boldsymbol{y}_{space} \in \mathbb{R}^d$ and $x_{time}, y_{time} \in \mathbb{R}$.

Definition 1

Let $\langle \cdot, \cdot \rangle$ is Euclidean inner product and $\langle \cdot, \cdot \rangle_{\mathcal{L}}$ denote the Lorentzian inner product that is induced by the Riemannian metric of the Lorentz model. For two vectors $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^{d+1}$, it is computed as follows:

$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle_{\mathcal{L}} = \langle \boldsymbol{x}_{space}, \boldsymbol{y}_{space} \rangle - x_{time} \cdot y_{time}$$

The induced Lorentzian norm is $||\boldsymbol{x}||_{\mathcal{L}} = \sqrt{|\langle \boldsymbol{x}, \boldsymbol{x} \rangle_{\mathcal{L}}|}$.

Definition 2

The Lorentz model possessing a constant curvature $-c$ is defined as a following set of vectors:

$$\mathcal{L}^d = \left\{ \boldsymbol{x} \in \mathbb{R}^{d+1} : \langle \boldsymbol{x}, \boldsymbol{x} \rangle_{\mathcal{L}} = -1/c, c > 0 \right\}$$

Lifting embeddings onto the hyperboloid

- Let the embedding vector from text or image encoder be $\boldsymbol{v} \in \mathbb{R}^d$.
- Exponential map:

$$\boldsymbol{x} = \operatorname{expm}_{\mathbf{o}}(\boldsymbol{v}) = [\boldsymbol{x}_{space}, x_{time}],$$

  where

$$\boldsymbol{x}_{space} = \frac{\sinh(\sqrt{c}||\boldsymbol{v}||)}{\sqrt{c}||\boldsymbol{v}||}\boldsymbol{v}, \quad x_{time} = \sqrt{1/c + ||\boldsymbol{x}_{space}||^2_{\mathcal{L}}}$$

※ Preventing numerical overflow

- Numerical issue: Exponential map amplifies $||\boldsymbol{v}|| \approx \sqrt{d}$ into $e^{\sqrt{d}}$, causing overflow.
- Example: For $d = 512, c = 1 \Rightarrow ||x_{space}|| \approx 6.7 \times 10^{10}$.
- Solution: Stabilize by scaling all $\boldsymbol{v}$ with learnable scalars $\alpha_{\text{img}}, \alpha_{\text{txt}}$.

Contrastive loss and Entailment loss

- For text and image, let the embedding vectors obtained via the exponential map be $\boldsymbol{x}_i = [\boldsymbol{x}_{i,space}, x_{i,time}], \boldsymbol{y}_i = [\boldsymbol{y}_{i,space}, y_{i,time}] \in \mathbb{R}^{d+1}$ for $i = 1, \ldots, N$.
- **Contrastive Loss** : $\mathcal{L}_{cont} = \sum_{i=j} d_{\mathcal{L}}(\boldsymbol{x}_i, \boldsymbol{y}_j) - \sum_{i \neq j} d_{\mathcal{L}}(\boldsymbol{x}_i, \boldsymbol{y}_j)$
  - Lorentzian distance:

$$d_{\mathcal{L}}(\boldsymbol{x}, \boldsymbol{y}) = \sqrt{1/c} \cdot \cosh^{-1}(-c\langle \boldsymbol{x}, \boldsymbol{y} \rangle_{\mathcal{L}})$$

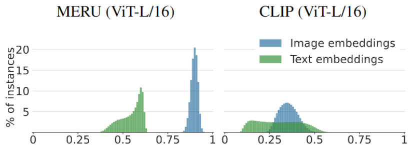
▶ **Entailment Loss** : $\mathcal{L}_{entail} = \sum_{i=1}^{N} \max\left(0, \text{ext}(\boldsymbol{x}_i, \boldsymbol{y}_i) - \text{aper}(\boldsymbol{x}_i)\right)$
  - Half-aperture:

$$\text{aper}(\boldsymbol{x}) = \sin^{-1}\left(\frac{2K}{\sqrt{c}\|x_{space}\|}\right), \quad K = 0.1.$$

  - Exterior angle (between $\boldsymbol{x}$ and paired $\boldsymbol{y}$):

$$\text{ext}(\boldsymbol{x}, \boldsymbol{y}) = \cos^{-1}\left(\frac{y_{time} + x_{time}\, c\langle \boldsymbol{x}, \boldsymbol{y} \rangle_{\mathcal{L}}}{\|\boldsymbol{x}_{space}\| \sqrt{(c\langle \boldsymbol{x}, \boldsymbol{y} \rangle_{\mathcal{L}})^2 - 1}}\right)$$

Experimental Results



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

Figure 4. **Distribution of embedding distances from `[ROOT]`:** We embed all 12M training images and text using trained MERU and CLIP. Note that precise distance is not necessary for this analysis, so we compute simple monotonic transformations of distances, $d(\mathbf{z})$. MERU embeds text closer to `[ROOT]` than images.

Table 1. **Zero-shot image and text retrieval.** Best performance in every column is highlighted in green. MERU performs better than CLIP for both datasets and across all model sizes.

		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**

Experimental Results

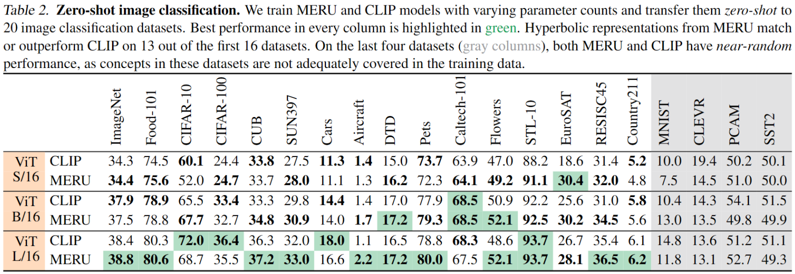


Table 2. **Zero-shot image classification.** We train MERU and CLIP models with varying parameter counts and transfer them *zero-shot* to 20 image classification datasets. Best performance in every column is highlighted in green. Hyperbolic representations from MERU match or outperform CLIP on 13 out of the first 16 datasets. On the last four datasets (gray columns), both MERU and CLIP have *near-random* performance, as concepts in these datasets are not adequately covered in the training data.

		ImageNet	Food-101	CIFAR-10	CIFAR-100	CUB	SUN397	Cars	Aircraft	DTD	Pets	Caltech-101	Flowers	STL-10	EuroSAT	RESISC45	Country211	MNIST	CLEVR	PCAM	SST2
ViT S/16	CLIP	34.3	74.5	**60.1**	24.4	**33.8**	27.5	**11.3**	**1.4**	15.0	**73.7**	63.9	47.0	88.2	18.6	31.4	**5.2**	10.0	19.4	50.2	50.1
	MERU	**34.4**	**75.6**	52.0	**24.7**	33.7	**28.0**	11.1	1.3	**16.2**	72.3	**64.1**	**49.2**	**91.1**	**30.4**	**32.0**	4.8	7.5	14.5	51.0	50.0
ViT B/16	CLIP	**37.9**	**78.9**	65.5	**33.4**	33.3	29.8	**14.4**	1.4	17.0	77.9	**68.5**	50.9	92.2	25.6	31.0	**5.8**	10.4	14.3	54.1	51.5
	MERU	37.5	78.8	**67.7**	32.7	**34.8**	**30.9**	14.0	**1.7**	**17.2**	**79.3**	**68.5**	**52.1**	**92.5**	**30.2**	**34.5**	5.6	13.0	13.5	49.8	49.9
ViT L/16	CLIP	38.4	80.3	**72.0**	**36.4**	36.3	32.0	**18.0**	1.1	16.5	78.8	**68.3**	48.6	**93.7**	26.7	35.4	6.1	14.8	13.6	51.2	51.1
	MERU	**38.8**	**80.6**	68.7	35.5	**37.2**	**33.0**	16.6	**2.2**	**17.2**	**80.0**	67.5	**52.1**	**93.7**	**28.1**	**36.5**	**6.2**	11.8	13.1	52.7	49.3

End