HYPERBOLIC IMAGE-TEXT REPRESENTATIONS

Karan Desai[1], **Maximilian Nickel**[2], **Tanmay Rajpurohit**,
Justin Johnson[1,2] & **Ramakrishna Vedantam**[3] *
[1] University of Michigan [2] Meta AI [3] New York University
kdexd@umich.edu

ABSTRACT

Visual and linguistic concepts naturally organize themselves in a hierarchy, where a textual concept "dog" entails all images that contain dogs. Despite being intuitive, current large-scale vision and language models such as CLIP (Radford et al., 2021) do not explicitly capture such hierarchy. We propose MERU, a contrastive model that yields hyperbolic representations of images and text. Hyperbolic spaces have suitable geometric properties to embed tree-like data, so MERU can better capture the underlying hierarchy in image-text data. Our results show that MERU learns a highly interpretable representation space while being competitive with CLIP's performance on multi-modal tasks like image classification and image-text retrieval.

1 INTRODUCTION

Visual-semantic hierarchy. It is commonly said that *'an image is worth a thousand words'* – consequently, images contain a lot more information than the sentences that describe them. For example, given Fig. 1 (middle image), one might describe it as *'a cat and a dog playing in the street'* or with a less specific sentence like *'exhausted doggo'* or *'so cute <3'*. These are not merely diverse descriptions but contain varying levels of detail about the underlying semantic contents of the image. As humans, we can reason about the relative detail in each caption, and can organize such concepts into a meaningful visual-semantic hierarchy (Vendrov et al., 2016), namely, *'exhausted doggo'* → *'a cat and a dog playing in the street'* → (Fig. 1 middle image). Providing multimodal models access to this inductive bias about vision and language has the potential to improve generalization (Radford et al., 2021), interpretability (Selvaraju et al., 2017) and enable better exploratory data analysis of large-scale datasets (Schuhmann et al., 2022; Radford et al., 2021).

Vision-language representation learning. Approaches such as CLIP (Radford et al., 2021) and ALIGN (Jia et al., 2021) have catalyzed a lot of recent progress in computer vision by showing that Transformer-based (Vaswani et al., 2017) models trained using large amounts of image-text data from the internet can yield transferable representations, and such models can perform *zero-shot* recognition and retrieval using natural language queries. All these models represent images and text as vectors in a high-dimensional Euclidean, affine space and normalize the embeddings to unit L^2 norm. However, such a choice of geometry can find it hard to capture the visual-semantic hierarchy.

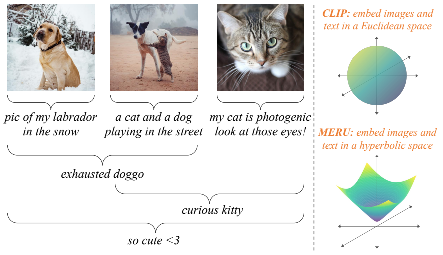


Figure 1: **Hyperbolic image-text representations. Left:** Images and text depict *concepts* and can be viewed in a *visual-semantic hierarchy*, where text *'exhausted doggo'* is more generic than image. **Right:** Representation manifolds of CLIP (*hypersphere*) and MERU (*hyperboloid*) illustrated in 3D. MERU assumes the origin to be the *most generic concept*, and embeds text closer to the origin than images.

An affine Euclidean space treats all embedded points in the same manner, with the same distance metric being applied to all points (Murphy, 2013). Conceptually, this can cause issues when modeling hierarchies – a *generic* concept (closer to the *root node* of the hierarchy) is close to many other concepts compared to a *specific* concept (which is only close to its immediate neighbors). Thus,

[*] KD and Rama did part of this work while at Meta.

a Euclidean space can find it hard to pack all the images that say a generic concept *'curious kitty'* should be close to while also respecting the embedding structure for *'a cat and a dog playing on the street'*. Such issues are handled naturally by hyperbolic spaces – the volume increases exponentially as we move away from the origin (Lee, 2019), making them a continuous relaxation of trees. This allows a generic concept (*'cat'*) to have many neighbors by placing it close to the origin (Nickel & Kiela, 2017), and more specific concepts further away. Thus, distinct specific concepts like images in Fig. 1 can be far away from each other while being close to some generic concept (*'animal'*).

Hyperbolic representations with MERU. In this work, we train the first large-scale contrastive image-text models that embed data in a hyperbolic representation space (Nickel & Kiela, 2017) – MERU that captures the visual-semantic hierarchy (Fig. 1). Importantly the hierarchy *emerges* in the representation space, given access only to image-text pairs during training such models. Practically, MERU confers multiple benefits such as (a) better performance on image retrieval and classification tasks, (b) more efficient usage of the embedding space, making it suited for resource-constrained, on-device scenarios, (c) an interpretable representation space that allows one to infer the relative semantic specificity of images and text. Overall, we summarize our contributions as follows:

- We introduce MERU, the first implementation of deep hyperbolic representations we are aware of, training ViTs (Dosovitskiy et al., 2021) with 12M image-text pairs.
- We provide a strong CLIP baseline that outperforms previous re-implementations (Mu et al., 2022) at comparable data scale, and systematically demonstrate the benefits of hyperbolic representations over this baseline on *zero-shot* retrieval and classification, and effectiveness for small embedding dimensions (Kusupati et al., 2022).
- We perform thorough qualitative analysis with MERU to demonstrate its potential for exploratory data analysis of large-scale multimodal datasets.

2 Approach

In this section, we discuss the modeling pipeline and learning objectives of MERU to learn hyperbolic representations of images and text. We use tools of hyperbolic geometry throughout our discussion, see Appendix B for a thorough discussion of the relevant topics.

Our model design is based on CLIP (Radford et al., 2021) due to its simplicity and scalability. As shown in Fig. 2a, we process images and text using two separate encoders, and obtain embedding vectors of a fixed dimension n. Beyond this step, we introduce two differences on CLIP: (1) instead of L2 normalization, we transfer the Euclidean embeddings from the encoder to the Lorentz hyperboloid, (2) we use the negative of geodesic distance in the contrastive loss, instead of cosine similarity.

We also use an additional textual entailment loss, illustrated for low-dimensions in Fig. 2b. See Appendix C for a detailed walkthrough of our model design and entailment loss.

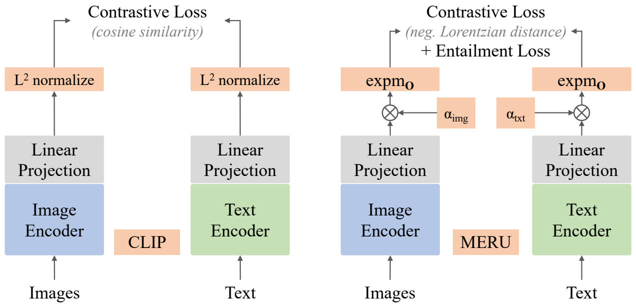


Figure 2(a): **Model design.** MERU shares similar architectural components as standard image-text contrastive models like CLIP. While CLIP projects the embeddings to hypersphere (via L^2 normalization), we lift them onto the Lorentz hyperboloid using the exponential map. We use the Lorentzian distance as a similarity metric in the contrastive loss, and use a special entailment loss to enforce *'text entails image'* partial order in the representation space.

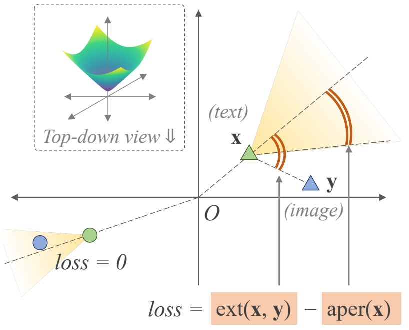


Figure 2(b): **Entailment loss in $\mathcal{L}^2$.** We enforce that an image embedding $\mathbf{y}$ lies inside a cone projected by the paired text embedding $\mathbf{x}$. This loss is implemented as the difference of exterior angle $\angle O\mathbf{xy}$ and half aperture of an imaginary cone at $\mathbf{x}$. Loss is zero if the image embedding is already inside the cone *(left quadrant)*.

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. MERU matches 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

3 EXPERIMENTS

Our main objective in the experiments is to establish the competitiveness of hyperbolic representations from our MERU models in comparison with their Euclidean counterparts. We also probe the trained models to assess the interpretability conferred by the hyperbolic structure.

Our primary comparison is with CLIP (Radford et al., 2021) which we re-implement and train using the RedCaps dataset (Desai et al., 2021). We train MERU models with the same training hyperparameters for fair and direct comparison. We train three models for CLIP and MERU, having Vision Transformers (Dosovitskiy et al., 2021) of varying capacity: ViT-S/B/L all with patch size 16. For quantitative evaluations, we perform *zero-shot* retrieval and classification as proposed by (Radford et al., 2021). See Appendix D for a description of training details and evaluation setup.

Table 1: **Zero-shot image and text retrieval.** Best performance (recall@5) in every column is highlighted in green. MERU performs better than CLIP for both datasets and across all model sizes.

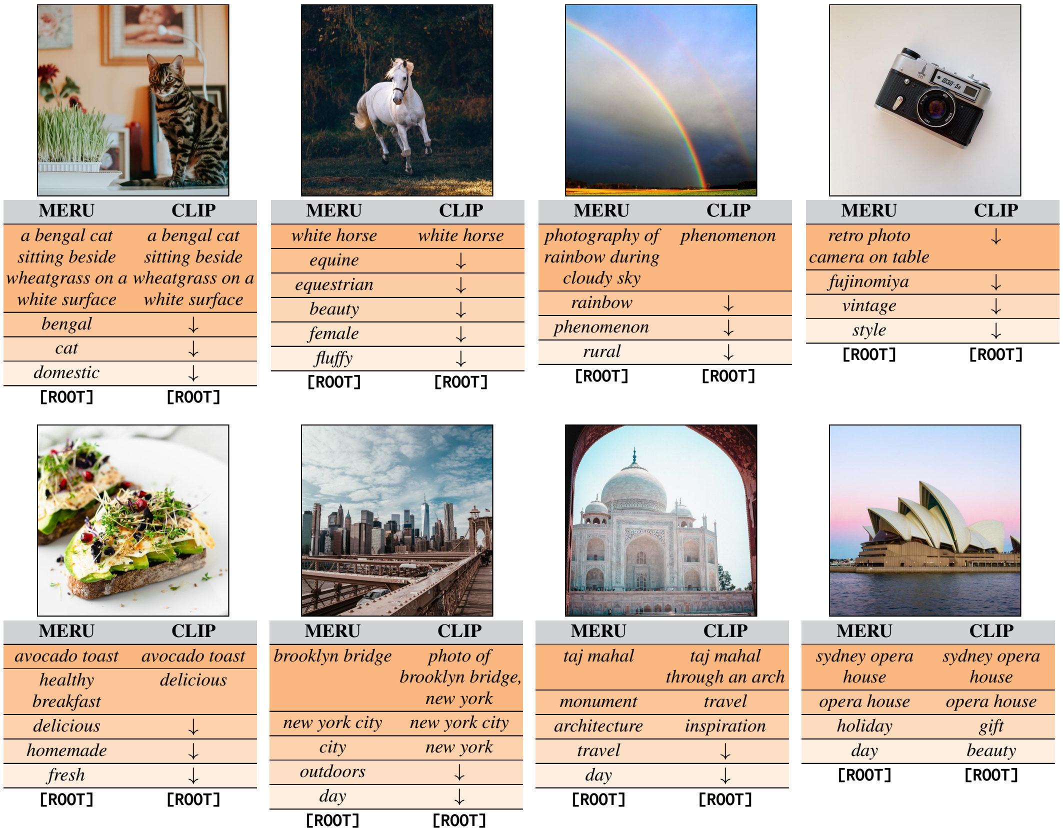
		text → image		*image → text*	
		COCO	Flickr	COCO	Flickr
ViT S/16	CLIP	29.9	35.3	37.5	42.1
	MERU	**30.5**	**37.1**	**39.0**	**43.5**
ViT B/16	CLIP	32.9	40.3	41.4	**50.2**
	MERU	**33.2**	**41.1**	**41.8**	48.1
ViT L/16	CLIP	31.7	39.0	40.6	47.8
	MERU	**32.6**	**39.6**	**41.9**	**50.3**

Image and text retrieval. Table 1 reports recall of MERU and the reproduced CLIP baselines on these benchmarks. Hyperbolic representations of MERU mostly perform best for all tasks and models (except Flickr30K text retrieval with ViT-B/16). This is encouraging evidence that hyperbolic spaces have suitable geometric properties to learn strong representations for retrieval applications. Surprisingly, increasing the model size to ViT-L/16 does not improve image retrieval for both, MERU and CLIP. We believe that better quality of text queries is required; increasing the size of text encoder can alleviate this issue.

Image classification. Table 2 shows strong transfer performance of MERU, matching or outperforming CLIP on 13 out of 16 standard datasets. While MERU is effective on recall-based measures (Table 1), it does not come at the expense of precision (Murphy, 2013). Overall, hyperbolic representations from MERU are competitive with their Euclidean counterparts across varying model architectures (ViT-S/B/L).

All models have *near-random* performance on four benchmarks. Concepts in these datasets have low coverage in RedCaps, like PCAM (Veeling et al., 2018) containing medical scans, or SST2 (Socher et al., 2013) containing movie reviews rendered as images. Performance on these benchmarks does not indicate the efficacy of our RedCaps-trained models; using larger training datasets like LAION (Schuhmann et al., 2022) may yield meaningful trends.

Qualitative analysis: Image traversals. In a discrete tree, one can discover the *ancestors* of any node by performing shortest-path traversal to the *root node* Dijkstra (1959). We perform such traversals for images with MERU and CLIP (ViT-L/16). If the representation space has captured the visual-semantic hierarchy, then a shortest-path traversal from an image to the embedding that represents the *most generic concept* (denoted as [ROOT]) should let us infer textual concepts that describe the image with varying levels of abstraction. Here we describe our analysis briefly, Appendix G for more details on estimating [ROOT] and additional results.



MERU	CLIP
a bengal cat sitting beside wheatgrass on a white surface	*a bengal cat sitting beside wheatgrass on a white surface*
bengal	↓
cat	↓
domestic	↓
[ROOT]	[ROOT]

MERU	CLIP
white horse	*white horse*
equine	↓
equestrian	↓
beauty	↓
female	↓
fluffy	↓
[ROOT]	[ROOT]

MERU	CLIP
photography of rainbow during cloudy sky	*phenomenon*
rainbow	↓
phenomenon	↓
rural	↓
[ROOT]	[ROOT]

MERU	CLIP
retro photo camera on table	↓
fujinomiya	↓
vintage	↓
style	↓
[ROOT]	[ROOT]

MERU	CLIP
avocado toast	*avocado toast*
healthy breakfast	*delicious*
delicious	↓
homemade	↓
fresh	↓
[ROOT]	[ROOT]

MERU	CLIP
brooklyn bridge	*photo of brooklyn bridge, new york*
new york city	*new york city*
city	*new york*
outdoors	↓
day	↓
[ROOT]	[ROOT]

MERU	CLIP
taj mahal	*taj mahal through an arch*
monument	*travel*
architecture	*inspiration*
travel	↓
day	↓
[ROOT]	[ROOT]

MERU	CLIP
sydney opera house	*sydney opera house*
opera house	*opera house*
holiday	*gift*
day	*beauty*
[ROOT]	[ROOT]

Figure 3: **Image traversals with MERU and CLIP.** We perform text retrieval at multiple steps while traversing from an image embedding to [ROOT]. Overall, CLIP retrieves fewer textual concepts (top row), but in some cases it reveals a coarse hierarchy (bottom row). MERU captures hierarchy with significantly greater detail, we observe that: (1) Text becomes more *generic* we move towards [ROOT], *e.g.*, *white horse → equestrian* and *retro photo camera → vintage*. (2) MERU has higher recall of concepts than CLIP, like words in bottom row: *homemade, city, monument*. (3) MERU also shows systematic text→image entailment, *e.g.*, *day* entails many images captured in daylight.

We traverse from an image and [ROOT] by interpolating 50 equally spaced steps along the geodesic connecting their embedding vectors. We use every interpolated step embedding as a query to perform retrieve the nearest neighbor from a set of text embeddings $\mathcal{X}$, that also include [ROOT]. We display results with 60 randomly selected images collected from pexels.com, a website that offers freely usable stock photos. We collect 750 captions from the associated image metadata on this website to create the set $\mathcal{X}$. Fig. 3 shows results with eight selected images and captions from pexels.com. CLIP seems to capture hierarchy to some extent, often retrieving very few (or zero) captions between image and [ROOT]. MERU captures it with finer granularity, retrieving concepts that gradually become more *generic* as we move closer to [ROOT].

4 Conclusion

In this paper, we learn large-scale image-text representations (MERU) to capture the visual-semantic hierarchy underlying images and text. Our key innovation is to bring the advances in learning hyperbolic representations to practical, large-scale deep learning applications. MERU is competitive or more performant than approaches that learn Euclidean representations (like CLIP) while also capturing hierarchical knowledge which allows one to make powerful inferences such as reasoning about images at different levels of abstraction, and performing semantic interpolations between images. Beyond this, our model also provides clear performance gains for small embedding dimensions (which are useful in resource-constrained settings). We hope this work catalyzes progress in learning useful representations from large amounts of unstructured data.

References

Mina Ghadimi Atigh, Julian Schoep, Erman Acar, Nanne van Noord, and Pascal Mettes. Hyperbolic Image Segmentation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 11, 20

Yushi Bai, Rex Ying, Hongyu Ren, and Jure Leskovec. Modeling Heterogeneous Hierarchies with Relation-specific Hyperbolic Cones. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 11

Jason Baumgartner, Savvas Zannettou, Brian Keegan, Megan Squire, and Jeremy Blackburn. The Pushshift Reddit Dataset. *arXiv preprint arXiv:2001.08435*, 2020. 19

Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – Mining Discriminative Components with Random Forests. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2014. 17, 19

Mert Bulent Sariyildiz, Julien Perez, and Diane Larlus. Learning visual representations with caption annotations. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2020. 11

Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 16

Tianqi Chen, Bing Xu, Chiyuan Zhang, and Carlos Guestrin. Training deep nets with sublinear memory cost. *arXiv preprint arXiv:1604.06174*, 2016. 19

Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. COCO captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015. 16

Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2021. 16

Gong Cheng, Junwei Han, and Xiaoqiang Lu. Remote Sensing Image Scene Classification: Benchmark and State of the Art. *Proceedings of the IEEE*, 2017. 17

Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing Textures in the Wild. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 17

Adam Coates, Andrew Ng, and Honglak Lee. An Analysis of Single Layer Networks in Unsupervised Feature Learning. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2011. https://cs.stanford.edu/~acoates/papers/coatesleeng_aistats_2011.pdf. 17

Shib Sankar Dasgupta, Michael Boratko, Dongxu Zhang, Luke Vilnis, Xiang Lorraine Li, and Andrew McCallum. Improving Local Identifiability in Probabilistic Box Embeddings. *arXiv preprint arXiv:2010.04831*, 2020. 11

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009. 17

Karan Desai and Justin Johnson. VirTex: Learning Visual Representations from Textual Annotations. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 11

Karan Desai, Gaurav Kaul, Zubin Aysola, and Justin Johnson. RedCaps: Web-curated image-text data created by the people, for the people. In *NeurIPS Datasets and Benchmarks*, 2021. 3, 16, 19

Edsger W Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1959. 3

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *arXiv preprint arXiv:2010.11929*, 2021. 2, 3, 11, 16

Albert Einstein. Zur Elektrodynamik bewegter Körper. *Annalen der physik*, 1905. 12

Albert Einstein, Hendrik A. Lorentz, Hermann Minkowski, and Hermann Weyl. *The Principle of Relativity: A Collection of Original Memoirs on the Special and General Theory of Relativity*. Martino Fine Books, 2nd edition, 2015. 12

Mohamed Elhoseiny, Babak Saleh, and Ahmed Elgammal. Write a Classifier: Zero-Shot Learning Using Purely Textual Descriptions. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2013. doi: 10.1109/ICCV.2013.321. 11, 17

Aleksandr Ermolov, Leyla Mirvakhabova, Valentin Khrulkov, Nicu Sebe, and Ivan Oseledets. Hyperbolic vision transformers: Combining improvements in metric learning. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 11

Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning Generative Visual Models from Few Training Examples: An Incremental Bayesian Approach Tested on 101 Object Categories. *CVPR Workshop*, 2004. 17

Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Marc Aurelio Ranzato, and Tomas Mikolov. Visual-Semantic Embedding Model. In *Advances in Neural Information Processing Systems (NeurIPS)*. Curran Associates, Inc., 2013. 11

Octavian-Eugen Ganea, Gary Bécigneul, and Thomas Hofmann. Hyperbolic Entailment Cones for Learning Hierarchical Embeddings. *arXiv preprint arXiv:1804.01882*, 2018. 11, 13, 14

Songwei Ge, Shlok Mishra, Simon Kornblith, Chun-Liang Li, and David Jacobs. Hyperbolic Contrastive Learning for Visual Representations beyond Objects. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 11

Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. Open-vocabulary Object Detection via Vision and Language Knowledge Distillation. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2022. 16

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 11

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 16

Patrick Helber, Benjamin Bischke, Andreas R. Dengel, and Damian Borth. EuroSAT: A Novel Dataset and Deep Learning Benchmark for Land Use and Land Cover Classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2019. 17

Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. OpenCLIP, 2021. URL https://doi.org/10.5281/zenodo.5143773. 16

Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling Up Visual and Vision-Language Representation Learning With Noisy Text Supervision. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2021. 1, 11

Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross B. Girshick. CLEVR: A Diagnostic Dataset for Compositional Language and Elementary Visual Reasoning. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 17

Andrej Karpathy and Li Fei-Fei. Deep Visual-Semantic Alignments for Generating Image Descriptions. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 11, 16

Valentin Khrulkov, Leyla Mirvakhabova, E. Ustinova, I. Oseledets, and Victor S. Lempitsky. Hyperbolic Image Embeddings. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 11, 20

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3D Object Representations for Fine-Grained Categorization. In *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*, 2013. 17

Alex Krizhevsky. Learning Multiple Layers of Features from Tiny Images. 2009. URL https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf. 17

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2012. 11

Aditya Kusupati, Gantavya Bhatt, Aniket Rege, Matthew Wallingford, Aditya Sinha, Vivek Ramanujan, William Howard-Snyder, Kaifeng Chen, Sham M. Kakade, Prateek Jain, and Ali Farhadi. Matryoshka Representation Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 2, 20

Marc Teva Law, Renjie Liao, Jake Snell, and Richard S. Zemel. Lorentzian Distance Learning for Hyperbolic Representations. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2019. 13

Matt Le, Stephen Roller, Laetitia Papaxanthos, Douwe Kiela, and Maximilian Nickel. Inferring Concept Hierarchies from Text Corpora via Hyperbolic Embeddings. *arXiv preprint arXiv:1902.00913*, 2019a. 11

Matthew Le, Stephen Roller, Laetitia Papaxanthos, Douwe Kiela, and Maximilian Nickel. Inferring Concept Hierarchies from Text Corpora via Hyperbolic Embeddings. In *Proceedings of the Annual Meeting on Association for Computational Linguistics (ACL)*, 2019b. 13

Yann LeCun, Corinna Cortes, and CJ Burges. MNIST handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2010. 17

John M. Lee. *Introduction to Riemannian Manifolds*. Graduate Texts in Mathematics. Springer International Publishing, 2019. ISBN 9783319917542. URL https://books.google.com/books?id=UIPltQEACAAJ. 2, 11, 15

Ang Li, Allan Jabri, Armand Joulin, and Laurens van der Maaten. Learning visual n-grams from web data. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2017. 11

Yangguang Li, Feng Liang, Lichen Zhao, Yufeng Cui, Wanli Ouyang, Jing Shao, Fengwei Yu, and Junjie Yan. Supervision Exists Everywhere: A Data Efficient Contrastive Language-Image Pretraining Paradigm. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2022. 16, 17

Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 16

Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019. 16

Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew B. Blaschko, and Andrea Vedaldi. Fine-Grained Visual Classification of Aircraft. *arXiv preprint arXiv:1306.5151*, 2013. 17

Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, et al. Mixed precision training. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018. 16, 19

George A Miller. WordNet: A Lexical Database for English. In *Speech and Natural Language: Proceedings of a Workshop Held at Harriman, New York*, 1992. URL https://aclanthology.org/H92-1116. 17

Hermann Minkowski. Raum und Zeit. *Physikalische Zeitschrift*, 1908. 12

Norman Mu, Alexander Kirillov, David Wagner, and Saining Xie. Slip: Self-supervision meets language-image pre-training. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2022. 2, 16, 17, 18, 19

Kevin P. Murphy. *Machine learning : a probabilistic perspective*. MIT Press, 2013. 1, 3

Kim Anh Nguyen, Maximilian Köper, Sabine Schulte im Walde, and Ngoc Thang Vu. Hierarchical Embeddings for Hypernymy Detection and Directionality. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2017. 11

Maximilian Nickel and Douwe Kiela. Poincaré Embeddings for Learning Hierarchical Representations. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. 2, 11

Maximilian Nickel and Douwe Kiela. Learning Continuous Hierarchies in the Lorentz Model of Hyperbolic Geometry. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2018. 13, 20

M-E. Nilsback and A. Zisserman. Automated Flower Classification over a Large Number of Classes. In *ICVGIP*, 2008. 17

Omkar Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and Dogs. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 17, 19

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 16

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning Transferable Visual Models From Natural Language Supervision. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2021. 1, 2, 3, 11, 13, 16, 17, 18, 19

John G. Ratcliffe. *Foundations of Hyperbolic Manifolds*. Graduate Texts in Mathematics. Springer New York, 2006. ISBN 9780387331973. URL https://books.google.com/books?id=JV9m8o-ok6YC. 11

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 2014. 20

Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *arXiv preprint arXiv:2111.02114*, 2021. 16

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. LAION-5B: An open large-scale dataset for training next generation image-text models. *arXiv preprint arXiv:2210.08402*, 2022. 1, 3, 16

Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2017. 1

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the Annual Meeting on Association for Computational Linguistics (ACL)*, 2016. 16

Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the Annual Meeting on Association for Computational Linguistics (ACL)*, 2018. 16

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, A. Ng, and Christopher Potts. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2013. 3, 17

Kihyuk Sohn. Improved Deep Metric Learning with Multi-class N-pair Loss Objective. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016. 11, 13

Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep Metric Learning via Lifted Structured Feature Embedding. *arXiv preprint arXiv:1511.06452*, 2015. 11

Bart Thomee, David A Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. YFCC100M: The New Data in Multimedia Research. *Communications of the ACM*, 2016. 16, 19

Alexandru Tifrea, Gary Bécigneul, and Octavian-Eugen Ganea. Poincaré GloVe: Hyperbolic Word Embeddings. *arXiv preprint arXiv:1810.06546*, 2018. 11

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2021. 16

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 (NeurIPS)*, 2017. 1, 16

Bastiaan S Veeling, Jasper Linmans, Jim Winkens, Taco Cohen, and Max Welling. CNNs for Digital Pathology. *arXiv preprint arXiv:1806.03962*, 2018. 3, 17

Ivan Vendrov, Ryan Kiros, Sanja Fidler, and Raquel Urtasun. Order-embeddings of images and language. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2016. 1, 11

Luke Vilnis, Xiang Lorraine Li, Shikhar Murty, and Andrew McCallum. Probabilistic Embedding of Knowledge Graphs with Box Lattice Measures. In *Proceedings of the Annual Meeting on Association for Computational Linguistics (ACL)*, 2018. 11

Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge J. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. 2011. 17

Ross Wightman. PyTorch Image Models. https://github.com/rwightman/pytorch-image-models, 2019. 16

Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 11

Jianxiong Xiao, James Hays, Krista A. Ehinger, Aude Oliva, and Antonio Torralba. SUN database: Large-scale scene recognition from abbey to zoo. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010. 17

Lewei Yao, Runhui Huang, Lu Hou, Guansong Lu, Minzhe Niu, Hang Xu, Xiaodan Liang, Zhenguo Li, Xin Jiang, and Chunjing Xu. FILIP: Fine-grained Interactive Language-Image Pre-Training. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2022. 16, 17

Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. In *Proceedings of the Annual Meeting on Association for Computational Linguistics (ACL)*, 2014. 11, 16

Alireza Zareian, Kevin Dela Rosa, Derek Hao Hu, and Shih-Fu Chang. Open-vocabulary object detection using captions. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 16

Xiaohua Zhai, Joan Puigcerver, Alexander Kolesnikov, Pierre Ruyssen, Carlos Riquelme, Mario Lucic, Josip Djolonga, André Susano Pinto, Maxim Neumann, Alexey Dosovitskiy, Lucas Beyer, Olivier Bachem, Michael Tschannen, Marcin Michalski, Olivier Bousquet, Sylvain Gelly, and Neil Houlsby. A Large-scale Study of Representation Learning with the Visual Task Adaptation Benchmark. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 17

Note on the method name: *Meru is a mountain that symbolizes the* center of all physical, metaphysical, and spiritual universes *in Eastern religions like Hinduism and Buddhism. Our method is named MERU because the origin of the hyperboloid entails everything and plays a more vital role than in Euclidean (or generally, affine) spaces. See also:* Mount Semeru, Indonesia *(Sources –* wikipedia.org/wiki/Mount_Meru *and* wikipedia.org/wiki/Semeru*)*

A Related work

Visual-language representation learning. Soon after the initial success of deep learning on ImageNet (Krizhevsky et al., 2012), deep metric learning (Sohn, 2016; Song et al., 2015) was used to learn vision-language representations in a shared semantic space (Frome et al., 2013; Karpathy & Fei-Fei, 2015). The motivations at the time included the possibility of improving vision models (Frome et al., 2013), enabling zero-shot learning by expressing novel categories as sentences (Frome et al., 2013; Elhoseiny et al., 2013), and better image-text retrieval (Karpathy & Fei-Fei, 2015; Young et al., 2014). Another line of work proposed learning visual models from language supervision via objectives like textual n-gram prediction (Li et al., 2017), or *generative* objectives like masked language modeling (Bulent Sariyildiz et al., 2020) or image captioning (Desai & Johnson, 2021).

More recent approaches like CLIP (Radford et al., 2021) and ALIGN (Jia et al., 2021) use contrastive metric learning to pre-train Vision Transformers (Dosovitskiy et al., 2021) and have helped to better realize the motivations of the earlier works in practice. While all prior works learn Euclidean embeddings, MERU explicitly works in the hyperbolic space that is conceptually better for embedding the visual-semantic hierarchy (Fig. 1) underlying images and text. Our results (Section 3) demonstrate that MERU yields strong performance as prior works, and also offers better interpretability to the representation space.

Entailment embeddings. In a vision and language context, Order Embeddings (Vendrov et al., 2016) propose capturing the partial order between language and vision by enforcing that text embeddings $\mathbf{x}$ and image embeddings $\mathbf{y}$, should satisfy $\mathbf{y} \leq \mathbf{x}$ for all dimensions i. While enforcing order is useful for retrieval, in our initial experiments, we found that distance-based contrastive learning to be crucial for better performance on classification and retrieval. Thus, we focus on adapting the currently successful contrastive learning and add our entailment objective in conjunction, to obtain the desired structure in the representation space.

For NLP and knowledge graph embedding applications, several approaches embed partially ordered data (Ganea et al., 2018; Nguyen et al., 2017; Bai et al., 2021; Dasgupta et al., 2020; Vilnis et al., 2018) or discover ordering from pairwise similarities (Tifrea et al., 2018; Nickel & Kiela, 2017; Le et al., 2019a). Our work has a flavor of both these lines of work, since we impose structure *across* modalities, but order also emerges *within* modality (Fig. 3).

Hyperbolic representations in computer vision. Khrulkov et al. (2020) learn hyperbolic image embeddings using image-label pairs, while Atigh et al. (2022) study image segmentation by utilizing hyperbolic geometry. More recently, Ermolov et al. (2022) and Ge et al. (2023) extend standard contrastive self-supervised learning framework (Wu et al., 2018; He et al., 2020) in vision to learn hyperbolic representations. In contrast to all these works, MERU learns multimodal representations with an order of magnitude more data and shows strong *zero-shot* transfer abilities across generic artificial intelligence tasks (Radford et al., 2021).

B Preliminaries

We briefly review Riemannian manifolds (Appendix B.1) and essential concepts of hyperbolic geometry (Appendix B.2). For a more thorough treatment of the topic, we refer the reader to textbooks by Ratcliffe (2006) and Lee (2019).

B.1 Riemannian manifolds

A *smooth surface* is a two-dimensional sheet which is *locally Euclidean* – every point on the surface has a local neighborhood which can be mapped to $\mathbb{R}^2$ via a differentiable and invertible function. *Smooth manifolds* extend the notion of smooth surfaces to higher dimensions.

A *Riemannian manifold* $(\mathcal{M}, g)$ is a smooth manifold $\mathcal{M}$ equipped with a *Riemannian metric* g. The metric g is a collection of inner product functions $g_{\mathbf{x}}$ for all points $\mathbf{x} \in \mathcal{M}$, and varies smoothly over the manifold. At any point $\mathbf{x}$, the inner product $g_{\mathbf{x}}$ is defined in the *tangent space* $\mathcal{T}_{\mathbf{x}}\mathcal{M}$, which is a Euclidean space that gives a linear approximation of $\mathcal{M}$ at $\mathbf{x}$. Euclidean space $\mathbb{R}^n$ is also a Riemannian manifold, where g is the standard Euclidean inner product.

Our main topic of interest is hyperbolic spaces, which are Riemannian manifolds with *constant negative curvature*. They are fundamentally different from Euclidean spaces that are *flat* (zero curvature). A hyperbolic manifold of n dimensions cannot be represented with $\mathbb{R}^n$ in a way that preserves both distances and angles. There are five popular models of hyperbolic geometry that either represent n-dimensional hyperbolic spaces either in $\mathbb{R}^n$ while distorting distances and/or angles (e.g. Poincaré ball model), or as a sub-manifold of $\mathbb{R}^{n+1}$ (e.g. the Lorentz model). We use the Lorentz model of hyperbolic geometry for developing MERU, which we briefly discuss next.

B.2 Lorentz model of hyperbolic geometry

The Lorentz model represents a hyperbolic space of n dimensions on the upper half of a two-sheeted hyperboloid in $\mathbb{R}^{n+1}$. See Fig. 1 for an illustration of $\mathcal{L}^2$ in $\mathbb{R}^3$. Hyperbolic geometry has a direct connection to the study of special relativity theory (Einstein, 1905; Einstein et al., 2015). We borrow some of its terminology in our discussion – we refer to the hyperboloid's axis of symmetry as *time dimension* and all other axes collectively as *space dimensions* (Minkowski, 1908). Hence we can write every vector $\mathbf{x} \in \mathbb{R}^{n+1}$ as $[\mathbf{x}_{space}, x_{time}]$, where $\mathbf{x}_{space} \in \mathbb{R}^n$ and $x_{time} \in \mathbb{R}$.

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

$$\mathcal{L}^n = \{\mathbf{x} \in \mathbb{R}^{n+1} : \langle \mathbf{x}, \mathbf{x} \rangle_{\mathcal{L}} = -{}^1\!/_c, c > 0\} \tag{1}$$

where $\langle \cdot, \cdot \rangle_{\mathcal{L}}$ denotes the *Lorentzian inner product*. This inner product is induced by the Riemannian metric of Lorentz model. For two vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^{n+1}$, it is computed as follows:

$$\langle \mathbf{x}, \mathbf{y} \rangle_{\mathcal{L}} = \langle \mathbf{x}_{space}, \mathbf{y}_{space} \rangle - x_{time}\, y_{time} \tag{2}$$

Here, $\langle \cdot, \cdot \rangle$ denotes the Euclidean inner product. The induced *Lorentzian norm* is $\|\mathbf{x}\|_{\mathcal{L}} = \sqrt{|\langle \mathbf{x}, \mathbf{x} \rangle_{\mathcal{L}}|}$. Every point on the hyperboloid satisfies the following constraint:

$$x_{time} = \sqrt{{}^1\!/_c + \|\mathbf{x}_{space}\|^2} \tag{3}$$

Geodesics. A *geodesic* is the shortest path between two points on the manifold. Geodesics in the Lorentz model are curves traced by the intersection of the hyperboloid with hyperplanes passing through the origin of $\mathbb{R}^{n+1}$. The *Lorentzian distance* between two points $\mathbf{x}, \mathbf{y} \in \mathcal{L}^n$ is:

$$d_{\mathcal{L}}(\mathbf{x}, \mathbf{y}) = \sqrt{{}^1\!/_c} \cdot \cosh^{-1}(-c\, \langle \mathbf{x}, \mathbf{y} \rangle_{\mathcal{L}}) \tag{4}$$

Tangent space. The tangent space at some point $\mathbf{z} \in \mathcal{L}^n$ is a Euclidean space of vectors that are orthogonal to $\mathbf{z}$ according to the Lorentzian inner product:

$$\mathcal{T}_{\mathbf{z}}\mathcal{L}^n = \{\mathbf{v} \in \mathbb{R}^{n+1} : \langle \mathbf{z}, \mathbf{v} \rangle_{\mathcal{L}} = 0\} \tag{5}$$

Any vector $\mathbf{u} \in \mathbb{R}^{n+1}$ can be projected to the tangent space $\mathcal{T}_{\mathbf{z}}\mathcal{L}^n$ via an orthogonal projection:

$$\mathbf{v} = \text{proj}_{\mathbf{z}}(\mathbf{u}) = \mathbf{u} + c\, \mathbf{z}\, \langle \mathbf{z}, \mathbf{u} \rangle_{\mathcal{L}} \tag{6}$$

Exponential and logarithmic maps. The *exponential map* provides a way to map vectors from tangent spaces onto the manifold. For a point $\mathbf{z}$ on the hyperboloid, it is defined as $\text{expm}_{\mathbf{z}} : \mathcal{T}_{\mathbf{z}}\mathcal{L}^n \to \mathcal{L}^n$ with the expression:

$$\mathbf{x} = \text{expm}_{\mathbf{z}}(\mathbf{v}) = \cosh(\sqrt{c}\|\mathbf{v}\|_{\mathcal{L}})\, \mathbf{z} + \frac{\sinh(\sqrt{c}\|\mathbf{v}\|_{\mathcal{L}})}{\sqrt{c}\|\mathbf{v}\|_{\mathcal{L}}}\, \mathbf{v} \tag{7}$$

Intuitively the exponential map shows how $\mathcal{T}_x\mathcal{L}^n$ *folds* on the manifold. Its inverse is the *logarithmic map* ($\text{logm}_{\mathbf{z}} : \mathcal{L}^n \to \mathcal{T}_{\mathbf{z}}\mathcal{L}^n$), that recovers $\mathbf{v}$ in the tangent space:

$$\mathbf{v} = \text{logm}_{\mathbf{z}}(\mathbf{x}) = \frac{\cosh^{-1}(-c\, \langle \mathbf{z}, \mathbf{x} \rangle_{\mathcal{L}})}{\sqrt{(c\, \langle \mathbf{z}, \mathbf{x} \rangle_{\mathcal{L}})^2 - 1}}\, \text{proj}_{\mathbf{z}}(\mathbf{x}) \tag{8}$$

For our approach, we only consider these maps where $\mathbf{z}$ is the hyperboloid origin ($\mathbf{O} = [\mathbf{0}, \sqrt{{}^1\!/_c}]$).

C APPROACH DETAILS

Lifting embeddings onto the hyperboloid. Let the embedding vector from the image encoder or text encoder, after linear projection be $\mathbf{v}_{enc} \in \mathbb{R}^n$. We need to apply a transformation such that the resulting vector $\mathbf{x}$ lies on the Lorentz hyperboloid $\mathcal{L}^n$ in $\mathbb{R}^{n+1}$.

Let the vector $\mathbf{v} = [\mathbf{v}_{enc}, 0] \in \mathbb{R}^{n+1}$ lie in the tangent space at the hyperboloid origin ($\mathbf{O} = [\mathbf{0}, \sqrt{1/c}]$ and $\mathbf{0} \in \mathbb{R}^n$). Note that $\mathbf{v}$ belongs to the tangent space, as the orthogonality condition (Eqn. (5)) is satisfied: $\langle \mathbf{O}, \mathbf{v} \rangle_{\mathcal{L}} = 0$. Thus, we parameterize *only* the *space* components of the Lorentz model ($\mathbf{v}_{enc} = \mathbf{v}_{space}$ henceforth). We then apply the exponential map $\mathrm{expm}_{\mathbf{O}}$, which simplifies as:

$$\mathbf{x} = \mathrm{expm}_{\mathbf{O}}(\mathbf{v}); \;\; \mathbf{x}_{space} = \frac{\sinh(\sqrt{c}\|\mathbf{v}_{space}\|)}{\sqrt{c}\|\mathbf{v}_{space}\|}\mathbf{v}_{space} \tag{9}$$

Note that $\|\cdot\|$ above is the regular Euclidean norm. The corresponding *time* component x_{time} can be computed from $\mathbf{x}_{space}$ using Eqn. (3). The final representation from the encoders is $\mathbf{x}_{space} \in \mathbb{R}^n$, such that $\mathbf{x} = [\mathbf{x}_{space}, x_{time}] \in \mathcal{L}^n$. See Appendix C.3 subsequently for full derivation.

Our parameterization is simpler than previous work which parameterizes vectors in full ambient space $\mathbb{R}^{n+1}$ (Law et al., 2019; Nickel & Kiela, 2018; Le et al., 2019b). Since we parameterize only the *space* components $\mathbf{x}_{space}$, the resulting $\mathbf{x}$ *always* lies on the hyperboloid. This eliminates the need for an orthogonal projection (Eqn. (6)) and simplifies the expression of the exponential map.

Preventing numerical overflow. The exponential map scales $\mathbf{v}_{space}$ using an exponential operator. According to CLIP-style weight initialization, $\mathbf{v}_{space} \in \mathbb{R}^n$ would have an expected norm $= \sqrt{n}$. After exponential map, it becomes $e^{\sqrt{n}}$, which can be numerically large (*e.g.,* $n = 512$ and $c = 1$ gives $||\mathbf{x}_{space}|| \approx 6.7 \times 10^{10}$).

To fix this issue, we *scale* all vectors $\mathbf{v}_{space}$ in a batch before applying $\mathrm{expm}_{\mathbf{O}}$ using two learnable scalars α_{img} and α_{txt}. These are initialized to $\sqrt{1/n}$ so that the Euclidean embeddings have an expected unit norm at initialization. We learn these scalars in logarithmic space to avoid collapsing all embeddings to zero. After training, they can be absorbed into the preceding projection layers.

Learning structured embeddings. Having lifted standard Euclidean embeddings onto the hyperboloid, we next discuss the losses used to enforce structure and semantics in representations learned by MERU. Recall that our motivation is to capture the visual-semantic hierarchy (Fig. 1) to better inform the generalization capabilities of vision-language models. For this, an important desideratum is a meaningful notion of distance between semantically similar text and image pairs. We also want to induce a partial order between text and images as per the visual-semantic hierarchy to have better interpretability. We do this with a modified version of an entailment loss proposed by Le et al. (2019b), that works for arbitrary hyperboloid curvatures $-c$.

C.1 CONTRASTIVE LEARNING FORMULATION

Given a batch of size B of image-text pairs and any j^{th} instance in batch, its image embedding $\mathbf{y}_j$ and text embedding $\mathbf{x}_j$ form a *positive* pair, whereas the remaining $B - 1$ text embeddings in the batch $\mathbf{x}_i (i \neq j)$ form *negative* pairs. In contrastive learning, we compute the negative Lorentzian distance as a similarity measure (Eqn. (4)) for all B pairs in the batch. These logits are divided by a temperature τ and apply a softmax operator. Similarly, we also consider a contrastive loss for text, that treats images as negatives. The total loss $\mathcal{L}_{cont}$ is the average of these two losses computed for every image-text pair in the batch. Our implementation of the contrastive loss is the same as the multi-class N-pair loss from (Sohn, 2016) used in CLIP (Radford et al., 2021) with the crucial difference being that we compute distances on the hyperboloid instead of cosine similarity.

C.2 ENTAILMENT LOSS

In addition to the contrastive loss, we adapt an entailment loss (Le et al., 2019b; Ganea et al., 2018) to enforce partial order relationships between paired text and images. Ganea et al. (2018) is more different from ours since they parameterize their representations according to the Poincaré ball model. Le et al. (2019b) use this loss with a fixed $c = 1$, which we extend to handle arbitrary curvatures.

Refer Fig. 2b for an illustration in two dimensions. Given text and image embeddings $\mathbf{x}$ and $\mathbf{y}$, we define an *entailment cone* for each $\mathbf{x}$, which narrows as we go farther from the origin. Note that the encoders only give $\mathbf{x}_{space}$ and $\mathbf{y}_{space}$ according to our parameterization. Corresponding x_{time} and y_{time}, whenever required, are calculated through Eqn. (3). This cone is defined by the half-aperture:

$$\text{aper}(\mathbf{x}) = \sin^{-1}\left(\frac{2K}{\sqrt{c}\;\|\mathbf{x}_{space}\|}\right) \tag{10}$$

where a constant $K = 0.1$ is used for setting boundary conditions near the origin. We now aim to identify and penalize when the paired image embedding $\mathbf{y}$ lies outside the entailment cone. For this, we measure the angle subtended by the arc from $\mathbf{y}$ to the axis of the entailment cone, shown as the exterior angle $\angle\mathbf{Oxy}$ in Fig. 2b:

$$\text{ext}(\mathbf{x}, \mathbf{y}) = \cos^{-1}\left(\frac{y_{time} + x_{time}\; c\;\langle\mathbf{x}, \mathbf{y}\rangle_{\mathcal{L}}}{\|\mathbf{x}_{space}\|\sqrt{\left(c\;\langle\mathbf{x}, \mathbf{y}\rangle_{\mathcal{L}}\right)^2 - 1}}\right) \tag{11}$$

If the exterior angle is smaller than the aperture, then the partial order relation between $\mathbf{x}$ and $\mathbf{y}$ is satisfied and we need not penalize anything, while if the angle is greater, we need to reduce it. This is captured by the following loss function (written below for one example $\mathbf{x}$, $\mathbf{y}$):

$$\mathcal{L}_{entail}(\mathbf{x}, \mathbf{y}) = \max(0,\; \text{ext}(\mathbf{x}, \mathbf{y}) - \text{aper}(\mathbf{x})) \tag{12}$$

We provide exact derivations of the above equations for half-aperture and exterior angle in Appendix C. Overall, our total loss is $\mathcal{L}_{cont} + \lambda\mathcal{L}_{entail}$ averaged over each minibatch.

C.3 Approach Derivations

Simplified exponential map. Recall that the output embedding from an encoder (image or text), $\mathbf{v}_{space} \in \mathbb{R}^n$ lies in a Euclidean space. Since $\mathcal{L}^n$ is embedded in $\mathbb{R}^{n+1}$, we defined $\mathbf{v} \in \mathbb{R}^{n+1}$ as $\mathbf{v} = [\mathbf{v}_{space}, 0]$ that belongs to the tangent space of the origin $\mathbf{O}$ of the hyperboloid. The exponential map can be written as Eqn. (7):

$$\mathbf{x} = [\mathbf{x}_{space}, x_{time}] = \text{expm}_{\mathbf{O}}([\mathbf{v}_{space}, 0]) \tag{13}$$

Recall that we only parameterize space components, and calculate time components using Eqn. (3) whenever needed. Hence, we resolve $\mathbf{x}_{space}$ from above as follows:

$$\mathbf{x}_{space} = \cosh(\sqrt{c}\;\|\mathbf{v}\|_{\mathcal{L}})\mathbf{0} + \frac{\sinh(\sqrt{c}\;\|\mathbf{v}\|_{\mathcal{L}})}{\sqrt{c}\;\|\mathbf{v}\|_{\mathcal{L}}}\mathbf{v}_{space} \tag{14}$$

First term reduces to $\mathbf{0}$. For $\mathbf{v} = [\mathbf{v}_{space}; 0]$ the Lorentzian norm simplifies to the Euclidean norm of *space* components:

$$\|\mathbf{v}\|^2_{\mathcal{L}} = \langle\mathbf{v}, \mathbf{v}\rangle_{\mathcal{L}} = \langle\mathbf{v}_{space}, \mathbf{v}_{space}\rangle - 0 \cdot 0 = \|\mathbf{v}_{space}\|^2$$

Substituting this in Eqn. (14) gives our simplified exponential map for *space* components used in the main paper (Eqn. (9)):

$$\mathbf{x}_{space} = \frac{sinh(\sqrt{c}\|\mathbf{v}\|)}{\sqrt{c}\|\mathbf{v}\|}\mathbf{v}_{space}$$

If one resolves the *time* component (x_{time}) in exponential map in Eqn. (13), they would arrive to the same value as substituting $\mathbf{x}_{space}$ from above in $x_{time} = \sqrt{1/c + \|\mathbf{x}_{space}\|^2}$.

Entailment loss: half-aperture. To derive the entailment loss for arbitrary curvatures $c > 0$, we start with the expression of half-aperture for a point $\mathbf{x}_b$ on the Poincaré ball as per Ganea et al. (2018):

$$\text{aper}_b(\mathbf{x}_b) = \sin^{-1}\left(K\frac{1 - c\,\|\mathbf{x}_b\|^2}{\sqrt{c}\,\|\mathbf{x}_b\|}\right) \tag{15}$$

The Poincaré ball model and Lorentz model are isometric to each other – one can transform any point from the Poincaré ball ($\mathbf{x}_b$) to the Lorentz model ($\mathbf{x}_h$) using this differentiable transformation:

$$\mathbf{x}_h = \frac{2\mathbf{x}_b}{1 - c\,\|\mathbf{x}_b\|^2} \tag{16}$$

The half-aperture of a cone should be invariant to the exact hyperbolic model we use, hence $\text{aper}_h(\mathbf{x}_h) = \text{aper}_b(\mathbf{x}_b)$. Substituting Eqn. (16) in Eqn. (15), we get the expression:

$$\text{aper}_h(\mathbf{x}_h) = \sin^{-1}\left(\frac{2K}{\sqrt{c}\,\|\mathbf{x}_h\|}\right)$$

Entailment loss: exterior angle. Consider three points $\mathbf{O}$ (the origin), $\mathbf{x}$ (text embedding) and $\mathbf{y}$ (image embedding). Then, a hyperbolic triangle is a closed shape formed by the geodesics connecting each pair of points. Similar to the Euclidean plane, the hyperbolic plane also has its law of cosines that allows us to talk about the angles in the triangle (Lee, 2019). Let the Lorentzian distances (Eqn. (4)) be $x = d(\mathbf{O}, \mathbf{y})$, $y = d(\mathbf{O}, \mathbf{x})$, and $z = d(\mathbf{x}, \mathbf{y})$. We can write the expression of *exterior angle* as follows:

$$\begin{aligned}\text{ext}(\mathbf{x}, \mathbf{y}) &= \pi - \angle\mathbf{Oxy} \\ &= \pi - \cos^{-1}\left[\frac{\cosh(z\sqrt{c})\cosh(y\sqrt{c}) - \cosh(x\sqrt{c})}{\sinh(z\sqrt{c})\sinh(y\sqrt{c})}\right]\end{aligned}$$

We use the relation $\pi - cos^{-1}(t) = cos^{-1}(-t)$ in the above equation. Then, let us define a function $g(t) = cosh(t\sqrt{c})$ for brevity, and substitute in the above equation. We also substitute $\sinh(t) = \sqrt{\cosh^2(t) - 1}$ as per the hyperbolic trigonometric identity. Putting it all together, we get:

$$\text{ext}(\mathbf{x}, \mathbf{y}) = \cos^{-1}\left[\frac{g(x) - g(z)g(y)}{\sqrt{g(z)^2 - 1}\sqrt{g(y)^2 - 1}}\right] \tag{17}$$

Now all we need is to compute $g(x)$, $g(y)$, and $g(z)$. We substitute the $z = d(\mathbf{x}, \mathbf{y})$ in $g(z)$ below:

$$\begin{aligned}g(z) &= \cosh\left(d(\mathbf{x}, \mathbf{y})\sqrt{c}\right) \\ &= \cosh\left(\frac{1}{\sqrt{c}}\cosh^{-1}(-c\,\langle\mathbf{x}, \mathbf{y}\rangle_{\mathcal{L}}) \cdot \sqrt{c}\right) \\ &= -c\,\langle\mathbf{x}, \mathbf{y}\rangle_{\mathcal{L}}\end{aligned}$$

Similarly, $g(x) = -c\langle\mathbf{O}, \mathbf{y}\rangle_{\mathcal{L}}$ and $g(y) = -c\langle\mathbf{O}, \mathbf{x}\rangle_{\mathcal{L}}$. The Lorentzian inner product (Eqn. (2)) with origin $\mathbf{O}$ simplifies:

$$\langle\mathbf{O}, \mathbf{x}\rangle_{\mathcal{L}} = -\frac{x_{time}}{\sqrt{c}} \quad \text{and} \quad \langle\mathbf{O}, \mathbf{y}\rangle_{\mathcal{L}} = -\frac{y_{time}}{\sqrt{c}}$$

Through this, we get $g(x) = x_{time}\sqrt{c}$ and $g(y) = y_{time}\sqrt{c}$. Finally, we can substitute $g(x)$, $g(y)$, and $g(z)$ to re-write Eqn. (17) to give the final expression as follows:

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

D Experimental design

In this section, we give a detailed description of how we train our MERU models and CLIP baselines, along with the evaluation protocol for all models presented in Section 3.

D.1 Training details

Baselines. We primarily compare with CLIP (Radford et al., 2021), that embeds images and text on a unit hypersphere in a Euclidean space. CLIP was trained using a private dataset of 400M image-text pairs. Several follow-up works re-implement CLIP and use publicly accessible datasets like YFCC (Thomee et al., 2016), Conceptual Captions (Sharma et al., 2018; Changpinyo et al., 2021), and LAION (Schuhmann et al., 2021; 2022); notable examples are OpenCLIP (Ilharco et al., 2021), SLIP (Mu et al., 2022), DeCLIP (Li et al., 2022), and FILIP (Yao et al., 2022). We develop our CLIP baseline and train it using a *single* public dataset – RedCaps (Desai et al., 2021) – for easier reproducibility. Our smallest model trains using 8× V100 GPUs in *less than one day* and significantly outperforms recent CLIP re-implementations that use YFCC (Mu et al., 2022) (Appendix E). Our implementation is based on PyTorch (Paszke et al., 2019) and `timm` Wightman (2019) libraries.

Models. We use the Vision Transformer (Dosovitskiy et al., 2021) as image encoder, considering three models of varying capacity – ViT-S (Touvron et al., 2021; Chen et al., 2021), ViT-B, and ViT-L. All use a patch size of 16. The text encoder is same as CLIP – a 12-layer, 512 dimensions wide Transformer (Vaswani et al., 2017) language model. We use the same byte-pair encoding tokenizer (Sennrich et al., 2016) as CLIP, and truncate input text at maximum 77 tokens.

Data augmentation. We randomly crop 50–100% area of images and resize them to 224×224, following (Mu et al., 2022). For text augmentation, we randomly *prefix* the subreddit names to captions as '`{subreddit} : {caption}`'.

Initialization. We initialize image/text encoders in the same style as CLIP, except for one change: we use a *sine-cosine* position embedding in ViT, like (Chen et al., 2021; He et al., 2022), and keep it frozen while training. We initialize the softmax temperature as $\tau = 0.07$ and clamp it to a minimum value of 0.01. For MERU, we initialize the learnable projection scalars $\alpha_{img} = \alpha_{txt} = {}^{1}/\sqrt{512}$, the curvature parameter $c = 1.0$ and clamp it in $[0.1, 10.0]$ to prevent training instability. All scalars are learned in logarithmic space as $log(1/\tau)$, $log(c)$, and $log(\alpha)$.

Optimization. We use AdamW (Loshchilov & Hutter, 2019) with weight decay 0.2 and $(\beta_1, \beta_2) = (0.9, 0.98)$. We disable weight decay for all gains, biases, and learnable scalars. All models are trained for $120K$ iterations with batch size 2048 ($\approx$ 20 epochs). The maximum learning rate is 5×10^{-4}, increased linearly for the first $4K$ iterations, followed by cosine decay to zero (Loshchilov & Hutter, 2016). We use mixed precision (Micikevicius et al., 2018) to accelerate training, except computing exponential map and losses for MERU in FP32 precision for numerical stability.

Loss multiplier (λ) for MERU. We set $\lambda = 0.2$ by running a hyperparameter sweep with ViT-B/16 models for one epoch. Some $\lambda > 0$ is necessary to induce partial order structure, however, quantitative performance is less sensitive to the choice of $\lambda \in [0.01, 0.3]$; Higher values of λ strongly regularize against the contrastive loss and hurt performance.

D.2 Evaluation setup

Image and text retrieval: CLIP-style models perform image and text retrieval within batch during training, making them ideal for retrieval-related downstream applications. We evaluate the retrieval capabilities of MERU as compared to CLIP on two established benchmarks: COCO and Flickr30K (Chen et al., 2015; Young et al., 2014), that comprise 5000 and 1000 images respectively and five captions per image. COCO evaluation uses the `val2017` split while Flickr30K uses the `test` split defined by Karpathy & Fei-Fei (2015). We perform *zero-shot transfer*, without any additional training.

We *squeeze* images to 224×224 pixels before processing them through the image encoder. For inference with MERU, we rank a pool of candidate image/text embeddings for retrieval in decreasing order of their Lorentzian inner product (Eqn. (2)) with a text/image query embedding. Some transfer tasks like *open-vocabulary detection* (Zareian et al., 2021; Gu et al., 2022) may require calibrated scores, for them we recommend using the training procedure – compute the negative of distance (Eqn. (4)), divide by temperature and apply a softmax classifier.

Image classification: Learning from language supervision allows CLIP to perform *zero-shot* image classification, wherein one may specify label sets as text queries (Elhoseiny et al., 2013) instead of using pre-defined ontologies (Miller, 1992; Deng et al., 2009). Classifier weights are obtained by embedding label-based queries (also called *prompts*) using the text encoder.

In the main paper, we evaluate MERU on 20 image classification benchmarks covering a wide variety of visual concepts. These are used by Radford et al. (2021) and several follow-up works (Mu et al., 2022; Yao et al., 2022; Li et al., 2022), and available with open-source libraries like `tensorflow-datasets` (`tensorflow.org/datasets`) and `torchvision` (`pytorch.org/vision`). We report top-1 mean per-class accuracy for all datasets to account for any label imbalance. We use multiple prompts per dataset, most of which follow Radford et al. (2021). We *ensemble* these multiple prompts by averaging their embeddings before lifting them onto the hyperboloid. See Tables 3 and 4 for details about datasets and prompts.

Table 3: **Datasets used for image classification evaluation.** Highlighted rows are datasets without an official validation split – we use a random held-out subset of the training split. EuroSAT and RESISC do not define any splits; we randomly sample non-overlapping splits. CLEVR Counts is derived from CLEVR (Johnson et al., 2017) and SST2 was introduced by (Socher et al., 2013).

Dataset	Classes	Train	Val	Test
Food-101 (Bossard et al., 2014)	101	68175	7575	25250
CIFAR-10 (Krizhevsky, 2009)	10	45000	5000	10000
CIFAR-100 (Krizhevsky, 2009)	100	45000	5000	10000
CUB-2011 (Wah et al., 2011)	200	4795	1199	5794
SUN397 (Xiao et al., 2010)	397	15880	3970	19849
Stanford Cars (Krause et al., 2013)	196	6515	1629	8041
FGVC Aircraft (Maji et al., 2013)	100	3334	3333	3333
DTD (Cimpoi et al., 2014)	47	1880	1880	1880
Oxf-IIIT Pets (Parkhi et al., 2012)	37	2944	736	3669
Caltech-101 (Fei-Fei et al., 2004)	102	2448	612	6084
Flowers (Nilsback & Zisserman, 2008)	102	1020	1020	6149
STL-10 (Coates et al., 2011)	10	4000	1000	8000
EuroSAT (Helber et al., 2019)	10	5000	5000	5000
RESISC (Cheng et al., 2017)	45	3150	3150	25200
Country211 (Radford et al., 2021)	211	31650	10550	21100
MNIST (LeCun et al., 2010)	10	48000	12000	10000
CLEVR Counts (Zhai et al., 2019)	8	4500	500	5000
PCAM (Veeling et al., 2018)	2	262144	32768	32768
SST2 (Radford et al., 2021)	2	6920	872	1821

Table 4: **Prompts used for zero-shot classification.** Most prompts are same as (Radford et al., 2021). We modify prompts for some datasets, that significantly improved performance for both MERU and CLIP – We did not extensively tune prompts but rather simply checked the performance on `val` splits for our CLIP baseline (Appendix E). **NOTE:** Some prompts use the word '`porn`' as it is included in the subreddit name. It does not indicate pornographic content but simply high-quality photographs.

ImageNet (our prompts)		
i took a picture : itap of a {}.	pics : a bad photo of the {}.	pics : a origami {}.
pics : a photo of the large {}.	pics : a {} in a video game.	pics : art of the {}.
pics : a photo of the small {}.		

Food-101 (our prompts)
food : {}.
food porn : {}.

CIFAR-10 and CIFAR-100
a photo of a {}.
a blurry photo of a {}.
a black and white photo of a {}.
a low contrast photo of a {}.
a high contrast photo of a {}.
a bad photo of a {}.
a good photo of a {}.
a photo of a small {}.
a photo of a big {}.
a photo of the {}.
a blurry photo of the {}.
a black and white photo of the {}.
a low contrast photo of the {}.
a high contrast photo of the {}.
a bad photo of the {}.
a good photo of the {}.
a photo of the small {}.
a photo of the big {}.

CUB-2011 (our prompts)
bird pics : {}.
birding : {}.
birds : {}.
bird photography : {}.

SUN397
a photo of a {}.
a photo of the {}.

Stanford Cars
a photo of a {}.
a photo of the {}.
a photo of my {}.
i love my {}!
a photo of my dirty {}.
a photo of my clean {}.
a photo of my new {}.
a photo of my old {}.

FGVC Aircraft
a photo of a {}, a type of aircraft.
a photo of the {}, a type of aircraft.

DTD (our prompts)
pics : {} texture.
pics : {} pattern.
pics : {} thing.
pics : this {} texture.
pics : this {} pattern.
pics : this {} thing.

Oxford-IIIT Pets
a photo of a {}, a type of pet.

Caltech-101
a photo of a {}.
a painting of a {}.
a plastic {}.
a sculpture of a {}.
a sketch of a {}.
a tattoo of a {}.
a toy {}.
a rendition of a {}.
a embroidered {}.
a cartoon {}.
a {} in a video game.
a plushie {}.
a origami {}.
art of a {}.
graffiti of a {}.
a drawing of a {}.
a doodle of a {}.
a photo of the {}.
a painting of the {}.
the plastic {}.
a sculpture of the {}.
a sketch of the {}.
a tattoo of the {}.
the toy {}.
a rendition of the {}.
the embroidered {}.
the cartoon {}.
the {} in a video game.
the plushie {}.
the origami {}.
art of the {}.
graffiti of the {}.
a drawing of the {}.
a doodle of the {}.

Oxford Flowers (our prompts)
flowers : {}.

STL10
a photo of a {}.
a photo of the {}.

EuroSAT
a centered satellite photo of {}.
a centered satellite photo of a {}.
a centered satellite photo of the {}.

RESISC
satellite imagery of {}.
aerial imagery of {}.
satellite photo of {}.
aerial photo of {}.
satellite view of {}.
aerial view of {}.
satellite imagery of a {}.
aerial imagery of a {}.
satellite photo of a {}.
aerial photo of a {}.
satellite view of a {}.
aerial view of a {}.
satellite imagery of the {}.
aerial imagery of the {}.
satellite photo of the {}.
aerial photo of the {}.
satellite view of the {}.
aerial view of the {}.

Country211
a photo i took in {}.
a photo i took while visiting {}.
a photo from my home country of {}.
a photo from my visit to {}.
a photo showing the country of {}.

MNIST
a photo of the number: "{}".

CLEVR
a photo of {} objects.

Patch Camelyon
this is a photo of {}.

Rendered SST2
a {} review of a movie.

E Developing a strong CLIP baseline

One of our contributions is to establish a lightweight, yet strong CLIP baseline. Original CLIP models (Radford et al., 2021) are trained using a private dataset of 400M image-text pairs across 128 GPUs for more than 10 days. We aim to maximize accessibility for future works, hence we decide our hyperparameters such that our smallest model can train on a single 8-GPU machine in under one day. We start with a reference CLIP ViT-S/16 baseline from SLIP (Mu et al., 2022) and carefully introduce one modification at a time. We benchmark improvements on zero-shot image classification

Table 5: **CLIP baseline.** We develop a strong CLIP baseline that trains on an 8-GPU machine in less than one day (ViT-S/16 image encoder), starting with SLIP (Mu et al., 2022) as a reference. We benchmark improvements on zero-shot image classification across 16 datasets. Our RedCaps-trained CLIP baseline (last row) is a significantly stronger baseline than its YFCC-trained counterparts.

	Images Seen	ImageNet	Food-101	CIFAR-10	CIFAR-100	CUB	SUN397	Cars	Aircraft	DTD	Pets	Caltech-101	Flowers	STL-10	EuroSAT	RESISC45	Country211	Average
YFCC15M-trained models																		
SLIP's CLIP (Mu et al., 2022)	368M	32.0	43.7	61.9	30.2	30.9	41.3	3.5	3.9	18.1	26.1	51.4	48.7	87.3	17.5	16.8	8.7	32.6
Our implementation	368M	33.1	42.3	64.9	34.4	33.7	43.8	2.9	5.1	19.1	25.0	49.8	47.2	87.4	26.8	21.6	9.0	34.1
+ BS 4096→2048	184M	28.2	34.2	58.7	29.4	27.4	39.4	2.9	4.3	16.5	20.1	43.8	42.2	85.4	20.2	19.0	8.5	30.0
+ *sin-cos pos embed*	184M	28.7	34.2	67.3	33.6	25.4	41.1	3.1	4.2	17.8	21.0	44.3	43.6	86.4	18.6	19.6	8.3	31.1
RedCaps-trained models																		
+ YFCC→RedCaps	184M	32.6	71.5	61.4	25.6	29.9	27.5	10.1	1.5	14.3	72.7	62.8	42.2	88.0	18.1	30.5	4.9	37.1
+ 90K→120K iters.	246M	33.9	72.5	60.1	24.4	30.0	27.5	11.3	1.4	13.1	73.7	63.9	44.4	88.2	18.6	31.4	5.2	37.5
+ our zero-shot prompts	246M	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	38.1

across 16 datasets used in our main experiments, using text prompts used by (Radford et al., 2021). Results are shown in Table 5.

CLIP baseline by SLIP. This re-implemented baseline was trained using a 15M subset of the YFCC dataset (Thomee et al., 2016). We re-evaluate the publicly released ViT-S/16 checkpoint [1] using our evaluation code; it obtains 32.6% average accuracy across all datasets.

Our re-implementation. We attempt a faithful replication of CLIP by following hyperparameters in SLIP. Our implementation obtains slightly higher average performance (34.1%) with three minor changes: (a) We use an *undetached* gather operation to collect all image/text features across all GPUs for contrastive loss. This ensures proper gradient flow across devices. (b) The above change allows using weight decay $= = 0.2$ like OpenAI's CLIP, unlike 0.5 used by SLIP's CLIP. (c) During training and inference, we resize input images using *bicubic* interpolation like original CLIP, instead of bilinear interpolation in SLIP's CLIP.

Fitting the model on 8-GPUs. This CLIP model requires 16× V100 32GB GPUs with a batch size of 4096 and automatic mixed precision (Micikevicius et al., 2018). Techniques like gradient checkpointing (Chen et al., 2016) can reduce memory requirements, but it comes at a cost of reduced training speed. Hence we avoid making it a requirement and simply reduce the batch size to 2048. This incurs a performance drop as the effective images seen by the model are halved. We offset the effective shortening of the training schedule by using fixed *sine-cosine* position embeddings in ViT, so learning position-related inductive biases is not required. This change slightly improves average accuracy (30.0% → 31.1% average accuracy).

Training with RedCaps dataset. RedCaps dataset (Desai et al., 2021) comprises 12M image-text pairs from Reddit, sourced from Pushshift (Baumgartner et al., 2020). Training with RedCaps significantly improves performance over YFCC-trained models (31.1% → 37.1% average accuracy), especially on datasets whose concepts have high coverage in RedCaps, *e.g.,* Food-101 (Bossard et al., 2014) and Pets (Parkhi et al., 2012).

To account for the smaller size of RedCaps, we increase the training iterations from 90K up to 120K. Finally, we modify zero-shot prompts for some datasets to match the linguistic style of RedCaps. For example, many captions in `r/food` simply mention the name of the dish in the corresponding image, hence we use the prompt '`food : {}`'. See Table 4 for the list of prompts for all datasets. We did not extensively tune these prompts, but we checked performance on the held-out validation sets to avoid cheating on the test splits. Finally, our CLIP ViT-S/16 baseline trains on 8× V100 32 GB GPUs within ≈14 hours and achieves 38.1% average performance across 16 datasets. We use these hyperparameters for all MERU and CLIP models in our experiments.

[1] `github.com/facebookresearch/slip`

F Additional Experiments

Table 6: Additional experiments and ablations with MERU.

(a) **MERU and CLIP with different embedding widths.** We report *zero-shot* COCO recall@5 and ImageNet top-1 accuracy. MERU outperforms CLIP at lower embedding widths.

		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**

(b) **MERU ablations.** We ablate three design choices of MERU and report *zero-shot* COCO recall@5 and ImageNet top-1 accuracy. Our design choices are crucial for training stability when using a larger model (ViT-L/16) with MERU.

	COCO *text→image*	COCO *image→text*	ImageNet
MERU ViT-B/16	33.2	41.8	37.5
1. *no entailment loss*	33.7	43.5	36.2
2. *fixed* $c = 1$	33.2	42.1	37.9
3. $\langle\cdot,\cdot\rangle_{\mathcal{L}}$ *in contrastive*	32.6	42.3	37.3
MERU ViT-L/16	32.6	41.9	38.8
1. *no entailment loss*	32.7	42.2	33.8
2. *fixed* $c = 1$	0.9	0.9	0.7
3. $\langle\cdot,\cdot\rangle_{\mathcal{L}}$ *in contrastive*	–	*did not converge*	–

Resource-constrained deployment. We hypothesize that embeddings that capture a rich visual-semantic hierarchy can use the volume in the representation space more efficiently. This is useful for on-device deployments with runtime or memory constraints that necessitate low-dimensional embeddings (Kusupati et al., 2022).

To verify this hypothesis, we train MERU and CLIP models that output 64–512 dimensions wide embeddings. We initialize the encoders from ViT-L/16 models (Table 2, last two rows) to reduce compute requirements, keep them frozen, and re-initialize projection layers and learnable scalars. We train for $30K$ iterations and evaluate on *zero-shot* COCO retrieval and ImageNet (Russakovsky et al., 2014) classification. Results in Table 6a show that MERU consistently performs better at low embedding widths. This indicates that hyperbolic embeddings may be an appealing solution for resource-constrained on-device applications.

Ablations. Next, we ablate our MERU models to observe the impact of our design choices. We experiment with two image encoders, ViT-B/16 and ViT-L/16, and evaluate for zero-shot COCO retrieval and ImageNet classification. Specifically, we train three ablations with the default hyperparameters (Appendix D.1), except having one difference each. Results are shown in Table 6b above.

- **No entailment loss:** We only use the contrastive loss for training this ablation. This effectively means setting $\lambda = 0$. Disabling the entailment loss is mostly inconsequential to MERU's performance. This shows that choosing a hyperbolic space is sufficient to improve *quantitative* performance over CLIP [2]. Entailment loss is crucial for better structure and interpretability, as seen in qualitative analysis.
- **Fixed curvature parameter:** Recall that our models treat the hyperboloid curvature as a learnable parameter during training. Here we train an ablation using a fixed curvature $c = 1$. This has negligible impact on MERU ViT-B/16, but learning curvature is crucial when scaling model size – MERU ViT-L/16 model with fixed $c = 1$ is difficult to optimize and performs poorly on convergence. As far as we are aware, no prior work learns the curvature (Atigh et al., 2022; Khrulkov et al., 2020; Nickel & Kiela, 2018).
- **Lorentzian inner product in contrastive loss:** CLIP-style contrastive loss uses the inner product defined on the hypersphere (cosine similarity). Similarly, we consider the *Lorentzian inner product* (Eqn. (2)) in the contrastive loss instead of negative Lorentzian distance. With this, MERU ViT-L/16 is difficult to train. Loss diverges due to numerical overflow, as Lorentzian inner product is numerically large and unbounded in $(-\infty, -{}^{1}\!/{}_{c}]$, unlike cosine similarity $\in [-1, 1]$. Lorentzian distance applies a logarithmic operator ($cosh^{-1}$) on the Lorentzian inner product, slowing down its growth and hence improving numerical stability.

[2] Note that this ablation is mathematically impossible for CLIP-style models as there is no obvious notion of entailment that can be defined when all the embeddings have a unit norm.

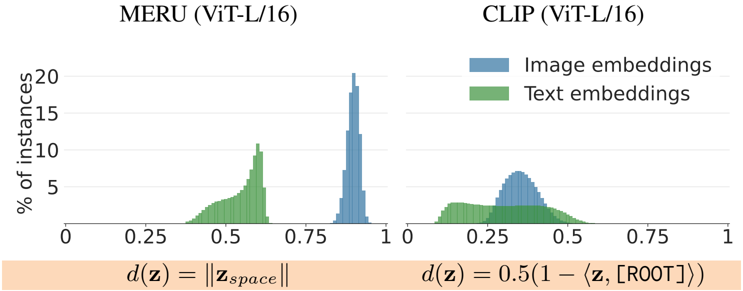


Figure 4: **Distribution of embedding distances from `[ROOT]`:** We embed all 12M training images and text using trained MERU and CLIP. Note that precise distance is not necessary for this analysis, so we compute simple monotonic transformations of distances, $d(\mathbf{z})$. MERU embeds text closer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

G Qualitative analysis

In this section, we expand more details on the qualitative analysis performed in the main paper Section 3. We probe our trained models to infer the visual-semantic hierarchy captured by MERU and CLIP. Apriori we hypothesize that MERU is better equipped to capture this hierarchy due to the geometric properties of hyperbolic spaces and an entailment loss that enforces the partial-order relationship *'text entails image'*. All our analysis in this section uses ViT-L/16 models.

Preliminary: `[ROOT]` embedding. Recall Fig. 1 – if we think of the visual-semantic hierarchy as a tree, then its *leaf nodes* are images and the *intermediate nodes* are text descriptions with varying *semantic specificity*. Naturally, the *root node* should represent the *most generic concept*. We denote its embedding in the representation space as `[ROOT]`.

For MERU, `[ROOT]` is the origin of the Lorentz hyperboloid as it entails the entire representation space. The location of `[ROOT]` for CLIP is not as intuitive – the notion of entailment is mathematically not defined, and the origin does not lie on the hypersphere. We empirically estimate CLIP's `[ROOT]` as an embedding vector that has the least distance from all embeddings of the training dataset. Hence, we average all 2×12M embeddings of images and text in RedCaps, followed by L^2 normalization. `[ROOT]` will be different for different CLIP models, whereas it is fixed for MERU.

Embedding distances from `[ROOT]`. In a representation space that effectively captures the visual-semantic hierarchy, text embeddings should lie closer to `[ROOT]` than image embeddings, since text is more *generic* than images (Fig. 1). To verify this, Fig. 4 shows the distribution of embedding distances from `[ROOT]` for MERU and CLIP. These distributions overlap for CLIP, but are separated for MERU. The range of distributions in Fig. 4 (left) hints that MERU embeds text and images in two *concentric, high-dimensional rings* around `[ROOT]`. The *ring* of text is more *spread out*, whereas ring of images is relatively *thin*. This mirrors the visual-semantic hierarchy – images only occupy *leaf nodes* whereas text occupies many intermediate nodes.

Additional results on image traversals. Figures in subsequent pages display results on the remaining images collected from `pexels.com` Every webpage on this website shows an image, a caption (often provided by the photographer), and additional tags (keywords) to search for similar images. We manually collect these captions and tags to create the set $\mathcal{X}$ of text embeddings. We perform parts-of-speech tagging on tags and only retain nouns and adjectives. Then, tags are converted to captions using prompts '`a photo of {}.`' for nouns, and '`this photo is {}.`' for adjectives.

Image sources. In the end, we provide a list of URLs of all images used in the paper. We display (and use) these images by taking a square crop. We thank all the photographers for generously sharing the images for free use.

MERU	CLIP
golden gate	golden gate bridge, san francisco, california
san francisco	famous landmark
tourist spot	↓
photo	↓
power	↓
[ROOT]	[ROOT]

MERU	CLIP
white cliffs of dover in england	white cliffs of dover
white cliffs of dover	cliffs
white	rocky
coast	↓
country	↓
[ROOT]	[ROOT]

MERU	CLIP
the famous fountain paint pots in yellowstone national park	yellowstone
yellowstone	beauty
national park	↓
power	↓
[ROOT]	[ROOT]

MERU	CLIP
the parthenon temple ruins in athens greece	the parthenon temple ruins in athens greece
historical site	famous landmark
architecture	low angle shot
domestic	↓
[ROOT]	[ROOT]

MERU	CLIP
big ben	big ben
holiday	↓
day	↓
[ROOT]	[ROOT]

MERU	CLIP
karlskirche	karlskirche church
architecture	church
style	↓
[ROOT]	[ROOT]

MERU	CLIP
fuji	fuji
japan	cozy
holiday	↓
[ROOT]	[ROOT]

MERU	CLIP
horseshoe bend	horseshoe bend
outdoors	national park
↓	credit
[ROOT]	[ROOT]

MERU	CLIP
milky way	↓
rural	↓
[ROOT]	[ROOT]

MERU	CLIP
volcano erupting at night under starry sky	volcano erupting at night under starry sky
active volcano	volcanic
outdoors	↓
[ROOT]	[ROOT]

MERU	CLIP
northern lights norway	northern lights norway
aurora	aurora
scenic	outdoors
outdoor	↓
[ROOT]	[ROOT]

MERU	CLIP
california	welcome to fabulous las vegas nevada signage
↓	famous landmark
[ROOT]	[ROOT]

MERU	CLIP
squirrel up on the snow covered tree	*squirrel up on the snow covered tree*
squirrel	*squirrel*
wildlife	↓
fluffy	↓
[ROOT]	[ROOT]

MERU	CLIP
seagull	*seagull*
bird	*bird*
air	↓
coast	↓
day	↓
[ROOT]	[ROOT]

MERU	CLIP
cute pug sitting on floor in white kitchen	*cute pug sitting on floor in white kitchen*
pug	↓
domestic	↓
little	↓
[ROOT]	[ROOT]

MERU	CLIP
three zebras	*three zebras*
zebras	*wild animals*
safari	↓
animal photography	↓
wild	↓
[ROOT]	[ROOT]

MERU	CLIP
monarch butterfly perching on red flower	*monarch butterfly*
monarch butterfly	↓
butterfly	↓
beauty	↓
day	↓
[ROOT]	[ROOT]

MERU	CLIP
red hibiscus in bloom	*red hibiscus in bloom*
hibiscus	*hibiscus*
bloom	*blooming flowers*
style	↓
[ROOT]	[ROOT]

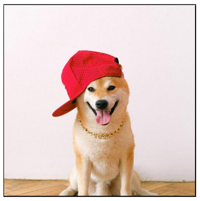
MERU	CLIP
white chicken on green grass field	*white chicken on green grass field*
cockerel	↓
chicken	↓
style	↓
[ROOT]	[ROOT]

MERU	CLIP
yellow blue and white macaw perched on brown tree branch	*yellow blue and white macaw perched on brown tree branch*
parrot	*parrot*
hungry	*animal*
female	↓
[ROOT]	[ROOT]

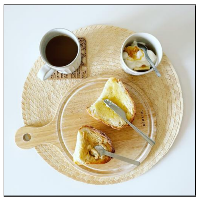
MERU	CLIP
edible agaric	*edible agaric*
mushroom	*mushroom*
beauty	*beauty*
little	↓
[ROOT]	[ROOT]

MERU	CLIP
aquatic animals	*aquatic animals*
sea life	*sea life*
style	*calamity*
[ROOT]	[ROOT]

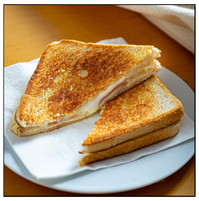
MERU	CLIP
an orca whale jumping out of the water	*an orca whale jumping out of the water*
whale	*whale*
[ROOT]	[ROOT]



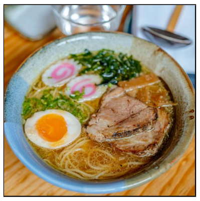
MERU	CLIP
financial	*adorable*
cute	↓
[ROOT]	[ROOT]



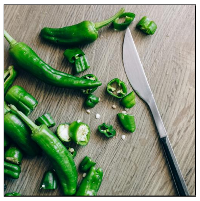
MERU	CLIP
bread and coffee for breakfast	*bread and coffee for breakfast*
pastry	↓
art	↓
[ROOT]	[ROOT]



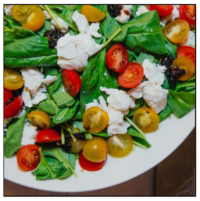
MERU	CLIP
grilled cheese	*grilled cheese*
lunch	↓
delicious	↓
classic	↓
[ROOT]	[ROOT]



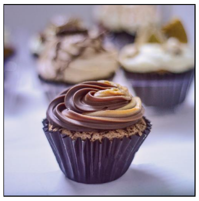
MERU	CLIP
bowl of ramen	*ramen*
local food	↓
tasty	↓
[ROOT]	[ROOT]



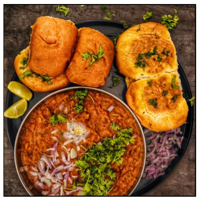
MERU	CLIP
green chili peppers and a knife	*green chili peppers and a knife*
spicy food	↓
spicy	↓
[ROOT]	[ROOT]



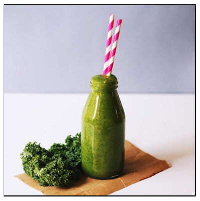
MERU	CLIP
spinach caprese salad	*spinach caprese salad*
lunch	*lunch*
homemade	↓
style	↓
[ROOT]	[ROOT]



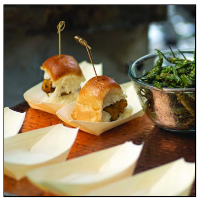
MERU	CLIP
cupcakes	*cupcakes*
chocolate cupcakes	↓
delicious	↓
homemade	↓
clean	↓
day	↓
[ROOT]	[ROOT]



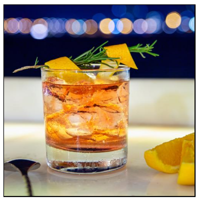
MERU	CLIP
pav bhaji	*pav bhaji dish on a bowl*
indian food	*indian food*
traditional food	*meal*
local food	*dinner*
spicy	↓
[ROOT]	[ROOT]



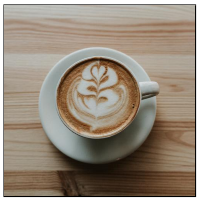
MERU	CLIP
clear glass bottle filled with broccoli shake	*smoothie*
smoothie	*homemade*
local food	*vegetable*
homemade	↓
spicy	↓
[ROOT]	[ROOT]



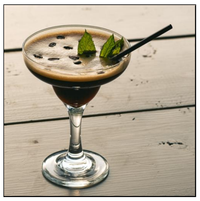
MERU	CLIP
vada pav	*cheese*
traditional food	↓
[ROOT]	[ROOT]



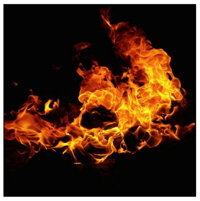
MERU	CLIP
old fashioned	*nutrition*
spicy	↓
style	↓
[ROOT]	[ROOT]



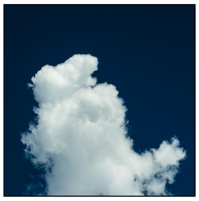
MERU	CLIP
latte	*latte*
design	↓
style	↓
[ROOT]	[ROOT]



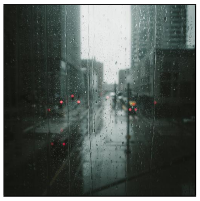
MERU	CLIP
espresso martini	↓
cocktail	↓
dessert	↓
hot	↓
[ROOT]	[ROOT]



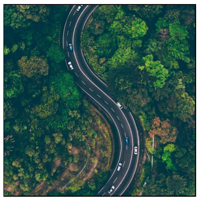
MERU	CLIP
campfire	*inferno*
fire	↓
blaze	↓
hot	↓
[ROOT]	[ROOT]



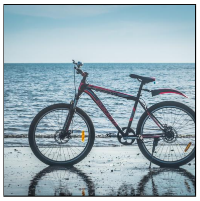
MERU	CLIP
cumulus	*cumulus*
white clouds	↓
clouds	↓
health	↓
fluffy	↓
[ROOT]	[ROOT]



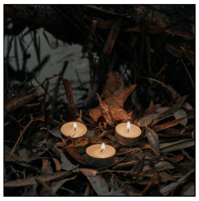
MERU	CLIP
raining in the city	*raining in the city*
weather	*downtown*
simple	↓
day	↓
[ROOT]	[ROOT]



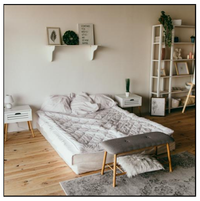
MERU	CLIP
road	*aerial view of road in the middle of trees*
travel	*aerial shot*
style	*rural*
↓	*clean*
[ROOT]	[ROOT]



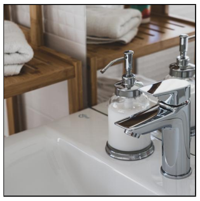
MERU	CLIP
mountain bike on the beach	*mountain bike on the beach*
analog	*bicycle*
retro	↓
style	↓
[ROOT]	[ROOT]



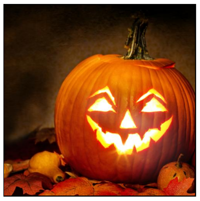
MERU	CLIP
lights	*white heart shaped candle on dried leaves*
evening	*holiday*
day	↓
[ROOT]	[ROOT]



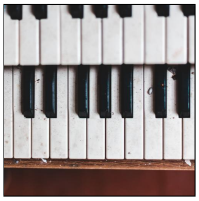
MERU	CLIP
bedroom	↓
clean	↓
[ROOT]	[ROOT]



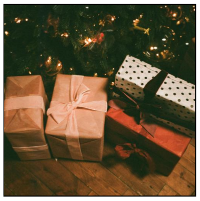
MERU	CLIP
clean bathroom	*stainless steel faucet on white ceramic sink*
investment	↓
clean	↓
[ROOT]	[ROOT]



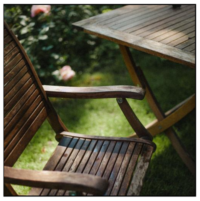
MERU	CLIP
jack o lantern with light	*jack o lantern with light*
carved pumpkin	↓
halloween	↓
hot	↓
[ROOT]	[ROOT]



MERU	CLIP
piano keys	*musical instrument*
keyboard	*music*
analog	↓
vintage	↓
style	↓
[ROOT]	[ROOT]



MERU	CLIP
assorted gift boxes on floor near christmas tree	*christmas presents*
christmas presents	*christmas gifts*
christmas gifts	↓
[ROOT]	[ROOT]



MERU	CLIP
garden table and chair	*seat*
table	↓
design	↓
comfort	↓
[ROOT]	[ROOT]

IMAGE CREDITS

1. pexels.com/photo/adult-yellow-labrador-retriever-standing-on-snow-field-1696589
2. pexels.com/photo/homeless-cat-fighting-with-dog-on-street-6601811
3. pexels.com/photo/short-coated-gray-cat-20787
4. pexels.com/photo/a-bengal-cat-sitting-beside-wheatgrass-on-a-white-surface-7123957
5. pexels.com/photo/white-horse-running-on-green-field-1996337
6. pexels.com/photo/photography-of-rainbow-during-cloudy-sky-757239
7. pexels.com/photo/retro-photo-camera-on-table-7162551
8. pexels.com/photo/avocado-toast-served-on-white-plate-10464867
9. pexels.com/photo/photo-of-brooklyn-bridge-new-york-2260783
10. pexels.com/photo/taj-mahal-through-an-arch-2413613
11. pexels.com/photo/sydney-opera-house-7088958
12. pexels.com/photo/antique-bills-business-cash-210600
13. pexels.com/photo/close-up-shot-of-a-cockatiel-13511241
14. pexels.com/photo/ripe-pineapple-on-gray-rock-beside-body-of-water-29555
15. pexels.com/photo/turned-on-floor-lamp-near-sofa-on-a-library-room-1907784
16. pexels.com/photo/golden-gate-bridge-san-francisco-california-1141853
17. pexels.com/photo/white-cliffs-of-dover-in-england-9692909
18. pexels.com/photo/the-famous-fountain-paint-pots-in-yellowstone-national-park-12767016
19. pexels.com/photo/the-parthenon-temple-ruins-in-athens-greece-14446783
20. pexels.com/photo/famous-big-ben-under-cloudy-sky-14434677
21. pexels.com/photo/karlskirche-church-7018621
22. pexels.com/photo/mt-fuji-3408353
23. pexels.com/photo/horseshoe-bend-arizona-2563733
24. pexels.com/photo/stars-at-night-1906667
25. pexels.com/photo/volcano-erupting-at-night-under-starry-sky-4220967
26. pexels.com/photo/northern-lights-1933319
27. pexels.com/photo/attraction-building-city-hotel-415999
28. pexels.com/photo/squirrel-up-on-the-snow-covered-tree-15306429
29. pexels.com/photo/a-seagull-flying-under-blue-sky-12509256
30. pexels.com/photo/cute-pug-sitting-on-floor-in-white-kitchen-11199295
31. pexels.com/photo/three-zebras-2118645
32. pexels.com/photo/monarch-butterfly-perching-on-red-flower-1557208
33. pexels.com/photo/red-hibiscus-in-bloom-5801054
34. pexels.com/photo/white-chicken-on-green-grass-field-58902
35. pexels.com/photo/yellow-blue-and-white-macaw-perched-on-brown-tree-branch-12715261
36. pexels.com/photo/closeup-photo-of-red-and-white-mushroom-757292
37. pexels.com/photo/photo-of-jellyfish-lot-underwater-3616240
38. pexels.com/photo/an-orca-whale-jumping-out-of-the-water-7767974
39. pexels.com/photo/yellow-labrador-retriever-wearing-red-cap-4588002
40. pexels.com/photo/bread-and-coffee-for-breakfast-15891938
41. pexels.com/photo/grilled-cheese-on-a-plate-14941252
42. pexels.com/photo/bowl-of-ramen-12984979
43. pexels.com/photo/green-chili-peppers-and-a-knife-5792428
44. pexels.com/photo/spinach-caprese-salad-on-white-ceramic-plate-4768996
45. pexels.com/photo/chocolate-cupcakes-635409
46. pexels.com/photo/pav-bhaji-dish-on-a-bowl-5410400
47. pexels.com/photo/clear-glass-bottle-filled-with-broccoli-shake-1346347
48. pexels.com/photo/vada-pav-15017417
49. pexels.com/photo/old-fashioned-cocktail-drink-4762719
50. pexels.com/photo/coffee-in-white-ceramic-teacup-on-white-ceramic-suacer-894696
51. pexels.com/photo/espresso-martini-in-close-up-photography-15082368
52. pexels.com/photo/photograph-of-a-burning-fire-672636
53. pexels.com/photo/white-clouds-in-blue-sky-8354530
54. pexels.com/photo/raining-in-the-city-2448749
55. pexels.com/photo/aerial-view-of-road-in-the-middle-of-trees-1173777
56. pexels.com/photo/mountain-bike-on-the-beach-10542237
57. pexels.com/photo/wax-candles-burning-on-ground-14184952
58. pexels.com/photo/white-wooden-shelf-beside-bed-2062431
59. pexels.com/photo/stainless-steel-faucet-on-white-ceramic-sink-3761560
60. pexels.com/photo/jack-o-lantern-with-light-5659699
61. pexels.com/photo/black-and-white-piano-keys-4077310
62. pexels.com/photo/assorted-gift-boxes-on-floor-near-christmas-tree-3394779
63. pexels.com/photo/garden-table-and-chair-14831985