

Language Supervision for Computer Vision

by

Karan P. Desai

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Computer Science and Engineering)
in the University of Michigan
2024

Doctoral Committee:

Assistant Professor Justin C. Johnson, Chair

Dr. Jason Baldridge, Google

Assistant Professor Andrew Owens

Professor Stella Yu

Karan P. Desai

kdexd@umich.edu

ORCID iD: [0009-0000-9739-3047](https://orcid.org/0009-0000-9739-3047)

© Karan P. Desai 2024

Acknowledgments

Before delving into the technical details of my research, I would like to take a moment and spread some *good vibes*. I am grateful to many people who have supported me and enriched my experience throughout my journey, this section is dedicated to them.

A huge thanks to Justin Johnson, for being an incredible advisor. Justin has provided me with deep technical guidance, while also offering the encouragement and freedom to boldly pursue new ideas and collaborations. I cherish all our discussions, ranging from ‘*Wow RedCaps crossed the 12 million mark? Amazing!*’ to ‘*Did you see NVIDIA announced the new Ampere series?*’, all the way to ‘*There is more to life than work, you should take as much time as you need for yourself*’. My experience has been joyful and overwhelmingly positive due to Justin, and I am looking forward to our future endeavors.

I thank the members of my doctoral committee – Jason Baldrige, Andrew Owens, and Stella Yu. To Jason, your valuable insights have been helpful to me, especially in the current landscape of industry research in my area of work. To Andrew and Stella, your critical feedback has pushed me to broaden the scope and extensions of my research. Beyond these discussions, I thank everyone for being generous with their time and patience.

Going forward, I take a walk down my memory lane and I thank all the humans that I have met and interacted with at different places and times in my journey.

Humans @ UMich (and Ann Arbor). I thank Mohamed El Banani and Nilesh Kulkarni for being wonderful collaborators, housemates, and supportive friends in these past five years. To Mohamed and Nilesh, I have enjoyed all the whiteboard discussions, meals, and memes. I hope that one day you can win against me at board games, good luck!

Special thanks to the students whom I had the pleasure to mentor during my PhD – Gaurav Kaul, Zubin Aysola, Shweta Singh, Aayan Yadav, Kemmannu Vineet Rao, and Kshama Shah. To you all, it has been my pleasure to watch you grow and advance in your career, I cannot wait to see what you do next! I also thank my external remote collaborators – Jitesh Jain, Ramprasaath Selvaraju and Nikhil Naik, for their valuable contributions and many hours of Zoom calls in the thick of the pandemic and later.

Heartfelt thanks to all my friends at UMich Vision and AI Lab – Chris Rockwell, Dandan Shan, Richard Higgins, Ayush Shrivastava, Ang Cao, Tiange Luo, Max Smith, James Boggs, Shengyi Qian, Linyi Jin, Daniel Geng, Ziyang Chen, Jeongsoo Park, Yiming Dou, Sarah Jabbour, Santiago Castro, Naihao Deng, Andrew Lee, and many more. I am sure our paths will cross in the future. I thank David Fouhey for all the helpful advice, hallway discussions, free coffee, and laughs! To Richard and Chris, you need to even out our rickroll score!

Navigating grad school would have been much harder without the support from many amazing humans across campus. I thank Brock Palen, Caleb Briggs, Matt Britt, and the entire team at Advanced Research Computing (ARC) for keeping *Great Lakes* (our GPU cluster) up and running smoothly. I also thank Ashley Andreae, Christina Certo, Karen Liska, Jasmin Stubblefield, and others in the CSE department staff for helping me with admin logistics and bureaucratic work. I thank Arahshiel Silver for reviewing my dissertation. I must also applaud the Counseling and Psychological Services (CAPS) team, and specifically Zubin DeVitre for helping me navigate through a phase of grief in my PhD.

I have grown to love working out of cafes in Ann Arbor – I thank the amazing staff of Songbird Cafe and Cannelle for their espresso and free Wi-Fi, which has fueled my research. And thanks to Ekdeep Singh Lubana, Puja Trivedi, and Gaurav Kaul, for being my frequent companions at these places. Also, a big shoutout to the amazing *Last Word!*

Thanks to the amazing Indian community for celebrating festivals and all the fun and laughs – Aayushi Shrivastava, Yash Trivedi, Harkirat Singh, Archi Agrawal, Janpreet Singh, Naman Saxena, and many others. Special shoutout to Kashmira Sawant for cooking top-tier *pav bhaji* and *vada pav*, and inviting me over for dinner!

I want to thank my persistent crew of board games and Dungeons & Dragons – Max Smith, Katelyn Boisvert, James Boggs, Chris Rockwell, Mohamed El Banani, Hayley Schroeder; friends at the *Heather* house (Matt Perez, Basia, Alex Kampiamba / Bubs), and the *Stool Crew* (Will, Kelsey, Avik, and Aidan). Board games have been a constant source of joy, and have ensured my weekly work-life balance.

Humans @ Meta. During my PhD, I did two summer internships at Meta – 2021 (remote), and 2022 (New York). I thank my managers during these internships – Ramakrishna Vedantam (Rama) and Laurens van der Maaten.

Rama has been an outstanding mentor, a long-term collaborator for the past five years, and a very reliable and supportive friend. Rama is a firm believer in getting back to the first principles – we discussed textbooks, sketched out proofs on whiteboard, and had in-depth discussions on engineering practices. Rama’s mentoring approach had a profound impact on how I approach my research. Laurens has shown great patience and has imparted

me with a positive attitude to encountering negative results in research. I also thank all my collaborators on the projects I did during these internships – Tanmay Rajpurohit, Maximilian Nickel, and Ishan Misra.

Thanks to all my peers at Meta for many stimulating discussions over countless meals – Karen Ullrich, Shubham Toshniwal, Mark Ibrahim, David Schwab, Rohit Girdhar, and many others. I cherish my evenings in the Meta office playing hundreds of games with the *Avalon Anonymous* group – Danielle Pintz, Kyle Urquhart, Ryan Tracy, Elliot Gorokhovsky, Stepan Hrudá, Goodwin Chen, James Cross, Kenneth Ng, Ravi Kodippili, Noam Brown, Rebecca Zhang, Satya Narayan Shukla, Jane Xu, Timothy Ram Pyari, and Klaudia Algiz. Thank you all, I will always be the *real Merlin*!

I had a very fun-loving and awesome cohort of co-interns in the summer of 2022, who I playfully call the *FAIR Coasters* – Andre Rubungo, Sagar Vaze, Desi Ivanova, Dejan Grubisic, Lyle Kim, Elizabeth Salesky, Andreea Oncescu, Pascal Sturmfels, Keren Fuentes, Mahi Shafiullah, Andrew Lee, David Liu, and Leo Adolphs. You all played a vital role in restoring the pre-pandemic normalcy in my life, thank you! It has been my pleasure to catch up with subsets of you periodically during conferences – let us keep doing that!

Last but not least, a special shoutout to my *Attaboyz* – Julius Berner and Steffen Schneider – for fruitful discussions over fruity cocktails, in New York, and later in London and New Orleans. Our *impacc* shall continue all over the world in the years to come!

Humans @ Georgia Tech. Before my PhD, I did a year-long research internship in the labs of Devi Parikh and Dhruv Batra, at the Georgia Institute of Technology. My journey as a researcher began here. To Dhruv and Devi, thank you for giving me the opportunity – I was an undergrad with no research background whatsoever, who sent you (what could be best described as) a *cold email* showing my enthusiasm and that I could sit down and code in Python. Most academics receive such requests in large amounts and often tend to ignore them; instead, Devi and Dhruv organized a fair recruitment process for me to showcase my potential, and later went above and beyond in clearing the administrative hurdles to hire me ¹. At all times, they have cared deeply to ensure my success.

I must thank Stefan Lee and Peter Anderson for a lot of hands-on guidance through my first research project (nocaps), and Harsh Agrawal for being a prolific co-author and friend. nocaps was an energy-packed and rewarding project, which made me think ‘*Let me do more of these*’. I thank all my collaborators on the two projects (nocaps, ProbNMN) I did while at Georgia Tech – Xinlei Chen, Rishabh Jain, Yufei Wang, Mark Johnson, Marcus

¹In US, it is challenging to host a foreign national as an intern after they graduate (me), they worked with GaTech admin to create a new job position that best describes my situation. That was quite something.

Rohrbach, and Ramakrishna Vedantam.

To all my wonderful friends in CVMLP (including those mentioned above), thank you for making that one year in Atlanta memorable and fun-filled – Ayush Shrivastava, Deshraj Yadav, Prithvijit Chattopadhyay, Nirbhay Modhe, Samyak Datta, Viraj Prabhu, Aishwarya Agrawal, Yash Goyal, Mohit Sharma, Purva Tendulkar, Arjun Chandrasekaran, Ashwin Kalyan, Erik Wijmans, Michael Cogswell, Jiasen Lu, and Jianwei Yang. Erik, Stefan, Yash, and Nirbhay – thank you for the board games and breadsticks! Folks at MStreet Apartments, our house parties were the best. Also thanks to Brenda Peters and Shubhangi Gupta, who are not officially a part of CVMLP but have been around for games and laughs!

This internship was a launchpad for my PhD – it instilled in me the self-confidence and resilience that helped me on my path forward.

Humans @ IIT Roorkee. I must thank all my friends from undergrad, who constantly cheer and support me and have been my source of comfort in times of distress. Thanks to the incredible *Ghissubros* – Saurabh Mishra, Mohit Virli, Nitin Jain, Tarannum Khan, Aditya Prakash, Hardik Chauhan, Hardik Jain, and Manu Agrawal. Heartfelt thanks to Punit Dhoot, Ketan Mittal, Anshul Shah, Purujit Goyal, and many of my friends in IITR CSE. I am incredibly thankful to have traveled around and met you all across the world during my PhD. It is surreal to think that I have known you all amazing people for almost a decade now, I look forward to us doing amazing things for many more years to come.

Humans @ Home. Last but not least, I thank my parents, Parul and Prakash Desai, for their unwavering support. They have made countless sacrifices toward my educational journey, including many during a phase of deep financial turmoil. I thank my extended family with a special shoutout to my maternal uncles, Bhadresh Desai, Ketan Desai, and Manoj Dube, for sparking my interest in books and computers since childhood.

In writing this dissertation, I pay tribute to the enduring legacy of my late uncle Satish Doshi, grandmother Shantaben Desai, and grandfather Amrutlal Desai. As I celebrate reaching this significant milestone, I can only imagine how proud they must be of my accomplishments. Their memory will always live on with me and my family.

Table of Contents

Acknowledgments	ii
List of Figures	viii
List of Tables	xiii
List of Appendices	xvi
Abstract	xvii
1 Introduction	1
2 Learning Visual Representations using Language	6
2.1 Introduction	6
2.2 Approach	8
2.3 Experiments	12
2.3.1 Linear probe evaluation	12
2.3.2 Fine-tuning based evaluation	16
2.3.3 Ablations	18
2.3.4 Image captioning	20
2.4 Related Work	21
2.5 Conclusion	24
3 Web-curated Image-Text Data from Reddit	25
3.1 Introduction	25
3.2 RedCaps: Collecting image-text pairs from Reddit	27
3.2.1 Data collection pipeline	28
3.2.2 Ethical considerations	29
3.3 RedCaps data analysis	31
3.4 Experiments	34
3.4.1 Transfer learning on downstream vision tasks	35
3.4.2 Image captioning	37
3.5 Related work	38
3.6 Conclusion	39
4 Hyperbolic Image-Text Representations	42
4.1 Introduction	42
4.2 Preliminaries	44
4.2.1 Riemannian manifolds	44
4.2.2 Lorentz model of hyperbolic geometry	45
4.3 Approach	46

4.3.1	Contrastive learning formulation	48
4.3.2	Entailment loss	48
4.4	Experiments	50
4.4.1	Training details	50
4.4.2	Image and text retrieval	51
4.4.3	Image classification	52
4.4.4	Resource-constrained deployment	54
4.4.5	Ablations	54
4.5	Qualitative analysis	56
4.5.1	Preliminary: Root node embedding	56
4.5.2	Embedding distances from the root node	57
4.5.3	Image traversals	57
4.6	Related work	64
4.7	Conclusion	65
5	How to Segment and Classify Anything?	67
5.1	Introduction	67
5.2	Related Work	69
5.3	Approach	70
5.3.1	Model architecture	70
5.3.2	Modeling Decisions	72
5.3.3	Training and Inference	73
5.4	Experiments	74
5.4.1	Zero-Shot Transfer	75
5.4.2	Transfer with Unlabeled Masks	78
5.4.3	Transfer with Labeled Masks	80
5.5	Conclusion	81
6	Conclusion	82
	Appendices	84
	Bibliography	136

List of Figures

2.1	Using language supervision for visual representation learning: We jointly train a ConvNet and Transformers using image-caption pairs, for the task of image captioning. Then, we transfer the learned ConvNet to vision tasks, for example, object detection.	6
2.2	Comparison of pre-training tasks for learning visual representations: Contrastive learning methods use a <i>semantically sparse</i> learning signal, encouraging different views of an image to have similar features. Image classification pairs an image with a single semantic concept, providing moderate semantic density. Multi-label classification, object detection, and instance segmentation increase semantic density by labeling and localizing multiple objects. Captions describe multiple objects, their attributes, relationships, and actions, giving a semantically dense learning signal. With VirTex, we aim to draw rich supervision from these semantically dense captions.	7
2.3	VirTex setup: Our model consists of a <i>visual backbone</i> (ResNet-50), and a <i>textual head</i> (two unidirectional Transformers). The visual backbone extracts image features, and textual head predicts captions via bidirectional language modeling (<i>bicaptioning</i>). The Transformers perform masked multiheaded self-attention over caption features, and multiheaded attention over image features. Our model is trained end-to-end from scratch. After pre-training, the visual backbone is transferred to downstream visual recognition tasks.	9
2.4	Data efficiency of VirTex: We compare VirTex and IN-sup models trained using varying amounts of images. VirTex closely matches or significantly outperforms IN-sup on downstream tasks despite using $10\times$ fewer images. ImageNet-supervised models using $\leq 10^5$ images are the mean of 5 trials, std dev. ≤ 1.0	14
2.5	Ablations: (i) Pre-training task. Bicaptioning improves over weaker pre-training tasks – forward captioning, token classification and masked language modeling. (ii) Visual backbone. Bigger visual backbones improve downstream performance – both, wider (R-50 w $2\times$) and deeper (R-101). (iii) Transformer size. Larger transformers (wider and deeper) improve downstream performance.	19

2.6	Image captioning with VirTex: We report the image captioning performance of VirTex models on COCO val2017 split, and some model-predicted captions. For the highlighted words, we visualize decoder attention weights from the textual head on the input image. Our model focuses on relevant image regions to predict objects (<i>shoes, desk</i>), background (<i>road</i>) as well as actions (<i>riding</i>).	21
2.7	We decode captions from the forward transformer of $L = 1, H = 512$ VirTex model using beam search. For the highlighted word, we visualize the decoder attention weights overlaid on the input image.	22
2.8	Attention visualizations per time step for predicted caption. We decode captions from the forward transformer of $L = 1, H = 512$ VirTex model using beam search. We normalize the attention masks to $[0, 1]$ to improve their contrast for better visibility.	23
3.1	RedCaps dataset comprises 12 million image-text pairs from 350 subreddits. RedCaps data contains everyday things that users like to share on social media, e.g., hobbies (r/crafts) and pets (r/shiba). Captions often contain specific and fine-grained descriptions (<i>northern cardinal, taj mahal</i>). Subreddit names provide image labels (r/shiba) even when captions may not (<i>mlem!</i>), and sometimes group many visually unrelated images through a common semantic meaning (r/perfectfit).	25
3.2	Preview of a Reddit image post: We collect the RedCaps dataset by downloading images and associated metadata (highlighted in orange) from Reddit image posts.	27
3.3	RedCaps was one of the largest public image-text datasets at the time of its creation. Unlike other datasets, it is expected to <i>grow</i> over time.	32
3.4	Top 20 subreddits with most image-text pairs in RedCaps.	32
3.5	RedCaps has a long-tailed distribution of caption lengths.	32
3.6	Number of unique words by POS, occurring at least 10 times (top), and frequent nouns in RedCaps (bottom).	33
3.7	Image captioning with VirTex-v2 trained on CC-3M vs RedCaps. Three crowd workers have observed these captions (without subreddit names) and voted the caption which seems more likely to be written by a human. The captions voted by majority of workers are underlined. Most of the voted captions are predicted by the RedCaps-trained model. These captions mention (top row): organic references (<i>little guy vs animal</i>), witty remarks (<i>snow sculpture</i>), and specific mentions (<i>singapore</i>).	40
3.8	Subreddit-controlled caption style. We prompt the VirTex-v2 model trained on RedCaps with subreddit names while decoding captions. We observe that such conditioning captures subtle linguistic structures (r/itookapicture: <i>itap of ..., r/somethingimade:</i> <i>i made...</i>). or changes the main subject of caption (r/earthporn: <i>venice</i> , r/food: <i>cold beer</i>). However, for completely unrelated images (saturn), the model tends to ignore the conditioning while generating captions.	41

4.1	Hyperbolic image-text representations. Left: Images and text depict <i>concepts</i> and can be jointly viewed in a <i>visual-semantic hierarchy</i> , wherein text ‘ <i>exhausted doggo</i> ’ is more generic than an image (which might have more details like a cat or snow). Our method MERU embeds images and text in a hyperbolic space that is well-suited to embed tree-like data. Right: Representation manifolds of CLIP (<i>hypersphere</i>) and MERU (<i>hyperboloid</i>) illustrated in 3D. MERU assumes the origin to represent the <i>most generic concept</i> , and embeds text closer to the origin than images.	42
4.2	MERU model design: MERU comprises similar architectural components as standard image-text contrastive models like CLIP. While CLIP projects the embeddings to a unit hypersphere, MERU lifts them onto the Lorentz hyperboloid using the exponential map. The contrastive loss uses the negative of Lorentzian distance as a similarity metric, and an entailment loss enforces ‘ <i>text entails image</i> ’ partial order in the representation space.	47
4.3	Entailment loss (illustrated for \mathcal{L}^2): This loss pushes image embedding y inside an imaginary cone projected by the paired text embedding x , and is implemented as the difference of exterior angle $\angle Oxy$ and half aperture of the cone. Loss is zero if the image embedding is already inside the cone (<i>left quadrant</i>).	49
4.4	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, $d(z)$. MERU embeds text closer to [ROOT] than images.	56
4.5	pexels.com webpage. We collect images and associated textual metadata (closed caption, <i>CC</i> and related keywords, ‘ <i>More like this</i> ’) from this website to create retrieval sets for the image traversal analysis.	58
4.6	Image traversals with MERU and CLIP. CLIP retrieves overall 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 <i>generic</i> we move towards [ROOT], <i>e.g.</i> , <i>white horse</i> → <i>equestrian</i> . (2) MERU has higher recall of concepts than CLIP, <i>e.g.</i> , <i>home-made</i> , <i>city</i> , <i>monument</i> . (3) MERU shows systematic text→image entailment, <i>e.g.</i> , <i>day</i> entails many images captured in daylight.	59
4.7	Image traversals with MERU and CLIP (locations and landmarks). Retrieved captions are sourced from pexels.com metadata. MERU captures a more systematic and fine-grained visual-semantic hierarchy than CLIP.	60
4.8	Image traversals with MERU and CLIP (flora and fauna). Retrieved captions are sourced from pexels.com metadata. MERU captures a more systematic and fine-grained visual-semantic hierarchy than CLIP.	61
4.9	Image traversals with MERU and CLIP (food and drinks). Retrieved captions are sourced from pexels.com metadata. MERU captures a more systematic and fine-grained visual-semantic hierarchy than CLIP.	62

4.10	Image traversals with MERU and CLIP (objects and scenes). Retrieved captions are sourced from pexels.com metadata. MERU captures a more systematic and fine-grained visual-semantic hierarchy than CLIP.	63
4.11	Image traversals (objects and scenes). Retrieved captions are sourced from pexels.com metadata. MERU captures a more systematic and fine-grained visual-semantic hierarchy than CLIP.	64
5.1	Zero-shot transfer with Segment and Classify Anything Model (SCAM). We introduce a general detector design composed of pre-trained vision models that <i>specialize</i> in segmentation (SAM) and classification (CLIP). Our careful design retains the capabilities of underlying models to enable fast and data-efficient transfer to object detection and instance segmentation. Figure shows high-scoring masks predicted by SCAM with CLIP ConvNeXt-XXL for <i>random</i> images from OpenImages [16] and LVIS [86] dataset. SCAM can segment novel objects without downstream fine-tuning up to the limit of pre-training knowledge in the constituent SAM and CLIP models.	67
5.2	SCAM Design: Image backbone extracts embeddings from an input image. The prompter uses backbone embeddings to propose a set of pixel locations for our visual concepts of interest. The segmenter uses the image and pixel prompts to predict a set of binary masks. Finally, the classifier performs mask classification using backbone embeddings and masks from the segmenter to predict class labels. Our backbone and segmenter are frozen throughout while we optionally train the light weight prompter and classifier for various data regimes.	71
5.3	Mask-based post-processing for SCAM. Left: We show predicted logits for the image from our prompter. Bright colors indicate pixels deep inside a mask, while darker regions are near or outside the object boundary. This allows us to sample a few high-quality point prompts for the segmenter. Right: SAM generates subpart masks that get classified as the full object. Since those masks wholly overlap with the full-object mask, they can be easily detected. We propose a sub-mask suppression (SMS) technique to efficiently find and suppress these masks.	75
5.4	Zero-shot results (qualitative). We observe that the baseline approach suffers from over-segmentation. Our test-time improvements with SCAM are effective in reducing these over-segmentations and producing reasonable outputs. <i>Note: Models receive RGB images, grayed here for better viewing.</i>	77
5.5	Transferring SCAM using unlabeled masks (low data regime). We observe the impact of training our prompter using subsets of COCO. We show that the prompter model is robust to the amount of data due to its lightweight design. The performance drop from 100% to 5% data is ≤ 1 AP.	79
5.6	Transferring SCAM using labeled masks. When trained with the same backbone, SCAM outperforms Mask R-CNN, with more prominent gains in a low-data regime.	80
B.1	Image traversals (1/20) using a set of captions from the YFCC dataset.	109

B.2	Image traversals (2/20) using a set of captions from the YFCC dataset.	110
B.3	Image traversals (3/20) using a set of captions from the YFCC dataset.	111
B.4	Image traversals (4/20) using a set of captions from the YFCC dataset.	112
B.5	Image traversals (5/20) using a set of captions from the YFCC dataset.	113
B.6	Image traversals (6/20) using a set of captions from the YFCC dataset.	114
B.7	Image traversals (7/20) using a set of captions from the YFCC dataset.	115
B.8	Image traversals (8/20) using a set of captions from the YFCC dataset.	116
B.9	Image traversals (9/20) using a set of captions from the YFCC dataset.	117
B.10	Image traversals (10/20) using a set of captions from the YFCC dataset.	118
B.11	Image traversals (11/20) using a set of captions from the YFCC dataset.	119
B.12	Image traversals (12/20) using a set of captions from the YFCC dataset.	120
B.13	Image traversals (13/20) using a set of captions from the YFCC dataset.	121
B.14	Image traversals (14/20) using a set of captions from the YFCC dataset.	122
B.15	Image traversals (15/20) using a set of captions from the YFCC dataset.	123
B.16	Image traversals (16/20) using a set of captions from the YFCC dataset.	124
B.17	Image traversals (17/20) using a set of captions from the YFCC dataset.	125
B.18	Image traversals (18/20) using a set of captions from the YFCC dataset.	126
B.19	Image traversals (19/20) using a set of captions from the YFCC dataset.	127
B.20	Image traversals (20/20) using a set of captions from the YFCC dataset.	128
C.1	Zero-shot transfer with SCAM: Qualitative results. High-scoring masks predicted by SCAM with CLIP ConvNeXt-XXL for <i>random</i> images from OpenImages [16]. SCAM can segment novel objects without <i>any</i> downstream fine-tuning.	134
C.2	Zero-shot transfer with SCAM: Qualitative results. High-scoring masks predicted by SCAM with CLIP ConvNeXt-XXL for <i>random</i> images from LVIS [86]. SCAM can segment novel objects without <i>any</i> downstream fine-tuning.	135

List of Tables

2.1	Annotation cost efficiency of VirTex: We compare the downstream performance of various pre-training methods on COCO. Cost estimates are reported in terms of <i>annotation worker hours</i> for COCO train2017 split. VirTex outperforms all other methods trained on the same set of images with the best performance vs. cost tradeoff.	13
2.2	System-level comparisons with VirTex: VirTex is competitive with recent SSL methods and concurrent work while using fewer images with language supervision.	15
2.3	Fine-tuning based evaluation of VirTex: We compare VirTex with different pre-training methods across four downstream tasks. For each task, all methods use the same architecture. We initialize the ResNet-50 backbone weights from pre-training (except Random Init), which are then fine-tuned end-to-end. Performance gaps with IN-sup are shown on the side. On all tasks, VirTex significantly outperforms all methods that use similar amount of pre-training images. VirTex closely matches or exceeds ImageNet supervised and self-supervised methods, despite using $10\times$ fewer pre-training images.	17
3.1	Automatic filtering: We use detectors to remove nearly 1.4M instances from RedCaps. We estimate the <i>precision</i> of these detectors by reviewing 5K random detected images. After filtering, we review 50K random images (out of 12M) to estimate <i>missed detections</i> , which we find to be very low. Caption filtering is deterministic (string matching).	30
3.2	Number of $\{1, 2, 3\}$ -grams occurring at least 10 times (top) and top-5 trigrams in each dataset (bottom).	33
3.3	Zero-shot image classification with VirTex-v2. We train models of exactly the same capacity using four different image-text datasets, then transfer them zero-shot to seven image classification datasets ($N = \#classes$).	35
3.4	Linear probe evaluation with VirTex-v2. We train logistic regression classifiers for seven image classification datasets, using frozen visual features extracted from models trained using four different image-text datasets.	37
3.5	Additional tasks: RedCaps trained model matches or exceeds models trained on SBU/CC-3M.	37

4.1	Zero-shot image and text retrieval. Best performance in every column is highlighted in green . MERU performs better than CLIP for both datasets and across all model sizes.	52
4.2	Zero-shot image classification. We train MERU and CLIP models with varying parameter counts and transfer them <i>zero-shot</i> to 20 image classification datasets. Best performance in every column is highlighted in <i>green</i> . Hyperbolic representations from MERU match or outperform CLIP on 13 out of the first 16 datasets. On the last four datasets (<i>gray</i> columns), both MERU and CLIP have <i>near-random</i> performance, as concepts in these datasets are not adequately covered in the training data.	53
4.3	MERU for resource-constrained deployment. We compare MERU and CLIP at different embedding widths on <i>zero-shot</i> classification and retrieval tasks (COCO recall@5 and ImageNet top-1 accuracy). MERU outperforms CLIP at lower embedding widths.	54
4.4	MERU ablations. We ablate three design choices of MERU and report <i>zero-shot</i> 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.	55
5.1	Zero-shot transfer with SCAM. We test experiment with our SCAM model three different datasets COCO [153], OpenImages [16] and LVIS [86]. The baseline method suffers from consistent errors due to background, and subparts. SCAM achieves substantial performance gains due to Mask NMS and subpart suppression across three different backbones all AP metrics. We show that with bigger backbones the performance improves. Running inference on SCAM and baselines in zero-shot mode requires ≈ 5 s per image.	76
5.2	Zero-shot transfer ablations. We ablate our design choices for SCAM on the COCO dataset [153]. Using our well-designed changes our method shows a perform improvement of 7.5 points over a naive standard baseline. We also show the upper bound segmentation performance of our design where we assume access to a perfect detector and prompt our model with GT bounding boxes.	78
5.3	Training with unlabeled masks. Transferring SCAM with unlabeled masks is beneficial for inference runtime ($5\times$ speedup). Training the prompter with unlabeled masks from the COCO dataset results in point prompts that are specific to the domain, and hence improves performance.	79
B.1	Evaluation datasets for MERU and CLIP. Datasets in highlighted rows do not have an official validation split – we use a random held-out subset of the training split. Underlined datasets do not define any splits; we randomly sample non-overlapping splits. *: <i>ImageNet is not used for linear probe evaluation so other splits are not necessary.</i>	102

B.2 **Prompts used for zero-shot image classification.** Most prompts are similar to Radford et al. [198] except for a few datasets on which we observed significant improvements for both MERU and CLIP using our custom prompts. 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. 103

B.3 **CLIP baseline.** We develop a strong CLIP baseline that trains on an 8-GPU machine in less than one day (ViT-S image encoder), starting with SLIP [176] 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. 105

B.4 **Linear probe evaluation.** We train a logistic regression classifier on embeddings extracted from the image encoders of CLIP and MERU (before projection layers). Note that embeddings from MERU are *not* lifted onto the hyperboloid. 106

C.1 **Zero-shot transfer to COCO, using different backbones with SCAM.** All models use our test-time improvements (Mask-based NMS, Sub-mask Suppression, and masked pooling). ConvNeXt-based models perform the best, which we use throughout our remaining experiments. 133

List of Appendices

A Web-curated Image-Text Data from Reddit	84
B Hyperbolic Image-Text Representations	100
C How to Segment and Classify Anything?	129

Abstract

Representation learning lies at the core of modern Artificial Intelligence. In computer vision, labeled image datasets like ImageNet have been the standard choice for representation learning. Despite being empirically successful, this approach is expensive to scale due to labeling costs. Moreover, the representation quality is limited by the size and diversity of datasets and their associated label ontologies.

My research explores using natural language supervision for computer vision. Using natural language allows us to go beyond fixed label ontologies and scale up to more general sources such as internet data. Toward this goal, my dissertation explores four problems – (1) Learning representations: I propose one of the first methods for language-supervised visual learning that uses image captioning as the training objective, showing its efficacy compared to ImageNet-trained methods on downstream tasks like object detection and segmentation. (2) Scaling data: I explore social media as a rich source of high-quality image descriptions and curate a dataset of 12 million image-text pairs while ensuring responsible curation practices. (3) Understanding data: It is difficult to comprehend the diversity of visual concepts present in millions of image-text pairs. I posit that images and text naturally organize into a tree-like hierarchy and propose an approach for learning representations that capture this hierarchy using tools from hyperbolic geometry. (4) Transfer to downstream tasks: Large vision-language models show impressive zero-shot transfer capabilities on image-level tasks like classification and retrieval. However, their transferability to pixel-level tasks like object detection and segmentation has relied on expensive labeled mask annotations. I propose an object detector to efficiently transfer pre-trained vision models to segment and classify visual objects without any fine-tuning, unlike existing detectors that train using orders of magnitude more labeled masks to achieve high performance.

In summary, my research affirms that using language supervision can drive the next leap of progress in computer vision and has immense utility in practical applications.

Chapter 1

Introduction

Computer vision is the pursuit of developing machine systems that can see and understand the world as humans do. Progress in computer vision is critical to the overarching goal of Artificial Intelligence (AI), as human-like perception and reasoning are essential components for general AI systems. Besides this broader goal, computer vision has countless practical applications assisting laypeople and domain experts alike, *e.g.*, autonomous vehicles, optical character recognition, image-based search engines, satellite imaging, and medical image segmentation, to name a few.

This dissertation work proposes general methods designed to adapt to such practical applications with minimal requirements of task-specific data and modeling solutions. Moreover, systems deployed to user-facing products must be robust, steerable, and interpretable. To this end, the central theme of this dissertation is to approach computer vision tasks by utilizing natural language supervision for visual inputs.

We begin by discussing the state of computer vision research before the undertaking of this dissertation to contextualize the presented work. Modern computer vision has its roots dating back to the early 1960s. The pioneering work of Roberts [205] paved the way for object recognition, illustrating the ability to extract three-dimensional shapes from images. The computational framework for visual information processing by Marr [169] is influential to the current theories of human perception. It is difficult to summarize the research progress of the last six decades for the scope of this chapter, thus we highlight the canonical developments of the previous decade.

Background

In the 2010s, the ImageNet dataset [47] has been paramount in shaping the landscape of computer vision. A popular subset of ImageNet containing 1.28 million images labeled with 1000 object categories has been the standard benchmark for visual recognition [208]. In 2012, a convolutional model named AlexNet [131] won the ImageNet competition, significantly outperforming prior methods using hand-crafted feature descriptors

like SIFT [163] and HOG [44]. This event propelled the mainstream adoption of deep learning techniques in computer vision, promoting the idea of data-driven *representation learning*. Stronger convolutional models followed gradually, e.g., VGG [217], Inception-Net [229, 230], ResNet [91, 261], Squeeze-Excitation [107], as well as general model families like EfficientNets [232] and RegNets [199]. The rapid growth of hardware computing power (GPUs) was a major catalyst for these architectural improvements.

Soon after the success of AlexNet, *transfer learning* became the de-facto approach to computer vision tasks – *pre-train* a deep convolutional network using the ImageNet dataset, then *fine-tune* it using a task-specific dataset. This recipe not only made advances on a myriad of downstream vision tasks like object detection [81], image segmentation [92, 160, 284], and monocular depth estimation [61], but also enabled new *multi-modal* tasks that would have otherwise required vast amounts of task-specific data, e.g., image captioning [249] and paragraph generation [127], visual question answering [6, 291], and vision-language navigation [5].

Aside from the architectural innovations for pre-training and downstream transfer, the community paid relatively little attention to the *unreasonable effectiveness of data* [227]. The ImageNet dataset remained the *consistent* pre-training data source, its size and diversity limit the quality of learned representations. Obtaining ImageNet-like data at scale is expensive due to the substantial cost of crowdsourcing *high-quality* image labels.

In subsequent years, there has been a growing interest in the search for scalable solutions for learning general visual representations, beyond the use of ImageNet. One approach is that of *weakly supervised learning* that uses internet images with low-quality noisy labels for representation learning. Examples include the JFT-300M dataset [102] (proprietary to Google) and its follow-up studies [40, 124, 227], and the Instagram dataset with billions of images labeled with hashtags [166, 265]. These studies observed that while low-quality image labels can facilitate representation learning, it is very *data-inefficient* – performance improvements on downstream transfer tasks are logarithmic upon increasing data by orders of magnitude.

Another approach is that of *self-supervised learning* (SSL), which tackles representation learning purely from unlabeled images. The core motivation of this strategy stems partly from cognitive science, noting that infants learn to identify and distinguish visual concepts long before knowing what they are called (*labels*). Besides, SSL offers an appealing benefit – being able to entirely side-step the labeling cost and leverage vast amounts of internet images. Representation learning without labels is done by performing auxiliary image-based tasks like context prediction [56], image colorization [282, 283], solving jigsaw puzzles [182], predicting image rotation [79], inpainting image patches [188], and so on.

A family of SSL methods is based on the contrastive learning framework [87] identifying two augmented views of an image, e.g., [32, 95, 257]. SSL methods yield strong transfer results when trained using unlabeled ImageNet images. However, right until the undertaking of this dissertation, these studies have been in an early phase and their scalability to billions of unlabeled images remains unexplored.

In summary, there has been an explosion of innovative methods for computer vision in the previous decade. Moving forward, the community grapples with open challenges – what is the right choice of data and learning methodology to break the transfer learning bottleneck and push the frontiers of computer vision? The time is ripe to explore scalable alternatives to the deeply entrenched ImageNet-based visual representation learning.

Motivation

This dissertation proposes using language supervision for computer vision as a promising path forward. Compared to *self-supervised* and *weakly-supervised* learning strategies discussed above, using natural language as a supervisory signal has appealing benefits.

- **Semantic density:** Natural language can convey rich semantic information carried by image pixels, with more descriptiveness than discrete label sets. We aim to enrich the supervisory signal for visual representation learning using language. Moreover, developing methods that can directly process vision-language data would let us side-step the necessity of crowdsourcing or curating image labels adhering to pre-defined ontology as required by ImageNet-supervised and weakly-supervised methods.
- **Data scalability:** Images are abundant on the internet ¹ and are often accompanied by some textual metadata. With language-supervised methods, we can leverage vast amounts of such internet data, instead of defining heuristics to convert the unstructured text into fixed label sets, or entirely discarding text and using only images like SSL.
- **Easy usage and adaptability:** Natural language is the standard mode of communication used by humans to describe and reason about the visual world. Models that can process vision-language inputs and produce outputs based on natural language instructions, can be easily promptable and steerable by users. Thus, such models can provide an intuitive user experience when deployed in real-world products.

We believe that language supervision for computer vision will unlock new and exciting applications that were nearly impossible with their predecessors, e.g. generating realistic images and videos based on text description.

¹It is important to note that internet data can have variable licensing and copyright terms which must be followed to curate data responsibly.

Thesis Statement and Contributions

Language supervision enables visual representation learning from scalable data sources, while yielding practical benefits such as interpretability and efficient adaptability to vision applications.

We support this statement by studying four essential problems in the overarching theme of this dissertation, which we discuss in the next four chapters.

Learning representations. In Chapter 2, we propose an approach for learning visual representations using language supervision, named **VirTex**. We trained VirTex models composed of convolutional networks (ResNets [91]) and Transformers [244] to perform the *generative* task of image captioning [249]. Under controlled comparisons, VirTex matched or exceeded equivalent ImageNet-trained models on six downstream vision tasks spanning image classification, object detection, and instance segmentation. This chapter is based on Desai and Johnson [50], published at **CVPR 2021**.

Scaling data. In Chapter 3, we aim to expand the scale and diversity of the training data for language-supervised visual learning, based on the empirical success of VirTex. Web data is abundant, albeit noisy and unstructured. Standard data curation practices involve using web crawlers, followed by extracting image URLs and HTML alt-text captions, *e.g.*, Conceptual Captions (CC [30, 215]). Complex filtering was crucial for quality control, which resulted in low-recall collection – CC filtered 5 billion images down to 3.3 million! We explore social media as a data source for high-recall curation without complex filtering. Our key intuition is that text on social media is of higher quality than HTML alt-text it is written with the intent of human interaction. We collected the **RedCaps** dataset comprising 12 million image-text pairs from Reddit, making it the largest dataset of its type at the time of release. We used RedCaps to train models for learning transferable visual representations and for image captioning (VirTex [50]). This chapter is based on Desai et al. [51] (published at **NeurIPS 2021**).

The growing scale of image datasets has prompted concerns pertaining to unwanted presence of gender stereotypes [24], NSFW imagery [18], and lack of geographic diversity [46]. With the development of RedCaps, we aim to propagate responsible collection practices for large image datasets – we filter images containing identifiable faces and NSFW content, and include an *opt-out* form on the dataset website to respect user privacy.

Understanding data. In Chapter 4, we aim to learn interpretable representations using image-text data. Large vision-language models have continued to scale to ever larger datasets soon after the undertaking of Chapter 3 – for example, the largest public image-text dataset in 2023 is LAION [210, 211] with more than 2 billion (English) image-text pairs, and used commonly to develop vision-language models like CLIP [198]. As data and models grow, they become increasingly opaque – the data diversity and complex relationships underlying billions of image-text pairs are difficult to comprehend. We posit that images and text naturally organize themselves in a hierarchy, where a textual concept (*e.g.*, ‘dog’) entails all images that depict a visual concept in different configurations (*e.g.*, dogs of various breeds, in various poses and scenes). Vision-language representations that capture such a hierarchy would let us infer higher-order relationships present in such datasets beyond their simple organization as independent image-text pairs. We introduce **MERU** [52], a contrastive image-text model that uses tools from hyperbolic geometry. We train MERU using RedCaps and obtain strong empirical results for *zero-shot* image classification and retrieval, spanning 18 datasets. For a given image, MERU can retrieve textual descriptions with varying levels of detail, showcasing the emergence of a *data-driven* hierarchy of concepts in the representation space.

Transfer to downstream tasks. Chapter 5 explores a new transfer learning recipe for the fundamental computer vision task of object detection. We have demonstrated the efficacy and scaling potential of language-supervised visual learning in the previous chapters – moving forward, we revisit the downstream modeling design for object detection which was largely based on the success of ImageNet-supervised pre-training. We propose revamping this recipe with the recent success in scaling two types of vision models: (a) Contrastive image-text models (*e.g.*, CLIP) that perform zero-shot image classification based on user-specified text prompts, and (b) Interactive segmentation models (*e.g.*, SAM [123]) that perform zero-shot segmentation based on user-specified points or box prompts. We compose these large vision models into a modular object detector that we name **SCAM**, short for Segment and Classify Anything Model. Our experiments compare SCAM with previous object detector designs like region-based models based on R-CNN [92, 151] and query-based models based on DETR [25] and Mask2Former [38]. Moreover, SCAM strongly retains the pre-training functionality of its constituent modules which allows its use for object detection in a training-free as well as low-data regime.

Chapter 2

Learning Visual Representations using Language

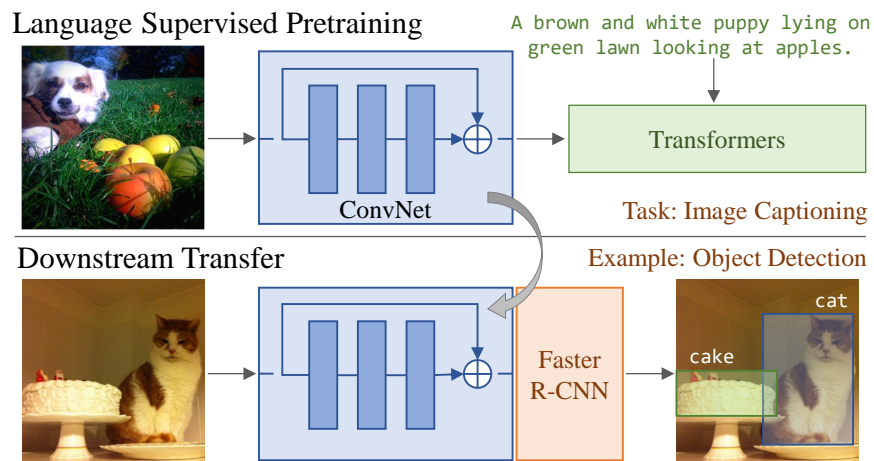


Figure 2.1: **Using language supervision for visual representation learning:** We jointly train a ConvNet and Transformers using image-caption pairs, for the task of image captioning. Then, we transfer the learned ConvNet to vision tasks, for example, object detection.

2.1 Introduction

The prevailing paradigm for learning visual representations is first to *pre-train* a convolutional network [91, 131] to perform image classification on ImageNet [47, 208], then *transfer* the learned features to downstream tasks [58, 214]. This approach has been wildly successful, and has led to significant advances in a wide variety of computer vision problems such as object detection [81], semantic [160] and instance [92] segmentation, image captioning [59, 117, 249], and visual question answering [6, 291]. Despite its practical success, this approach is expensive to scale since the pre-training step relies on images annotated by human workers.

For this reason, there has been increasing interest in *unsupervised pre-training* methods that use unlabeled images to learn visual representations which are then transferred to

users [48, 129]. In contrast, natural language descriptions do not require an explicit ontology and can easily be written by non-expert workers, leading to a simplified data collection pipeline [34, 104, 271]. Large quantities of weakly aligned images and text can also be obtained from internet images [168, 185, 215].

Our main contribution is to show that natural language can provide supervision for learning transferable visual representations with better data efficiency than other approaches. We train models from scratch on the COCO Captions dataset [34], and evaluate the learned features on downstream tasks including image classification, object detection, instance segmentation, and low-shot recognition. On all tasks, VirTex matches or exceeds the performance of existing methods for supervised or unsupervised pre-training on ImageNet, despite using up to $10\times$ fewer images. Our code and models are publicly available at github.com/kdexd/virtex.

2.2 Approach

Given a dataset of image-caption pairs, our goal is to learn visual representations that can be transferred to downstream visual recognition tasks. As shown in Figure 2.2, captions carry rich semantic information about images, including the presence of objects (*cat, plate, cake*); attributes of objects (*orange and white cat*); spatial arrangement of objects (*cat near a plate*); and their actions (*looking at apples*). Learned visual representations that capture such rich semantics should be useful for many downstream vision tasks.

To this end, we train *image captioning* models to predict captions from images. As shown in Figure 2.3, our model has two components: a *visual backbone* and a *textual head*. The visual backbone extracts visual features from an input image I . The textual head accepts these features and predicts a caption $C = (c_0, c_1, \dots, c_T, c_{T+1})$ token by token, where $c_0 = [\text{SOS}]$ and $c_{T+1} = [\text{EOS}]$ are fixed special tokens indicating the start and end of the sentence. The textual head performs bidirectional captioning (*bicaptioning*): it comprises a *forward model* that predicts tokens left-to-right, and a *backward model* that predicts right-to-left. All model components are randomly initialized, and jointly trained to maximize the log-likelihood of the correct caption tokens

$$\begin{aligned} \mathcal{L}(\theta, \phi) = & \sum_{t=1}^{T+1} \log \left(p(c_t \mid c_{0:t-1}, I; \phi_f, \theta) \right) \\ & + \sum_{t=0}^T \log \left(p(c_t \mid c_{t+1:T+1}, I; \phi_b, \theta) \right) \end{aligned} \tag{2.1}$$

where θ , ϕ_f , and ϕ_b are the parameters of the visual backbone, forward, and backward

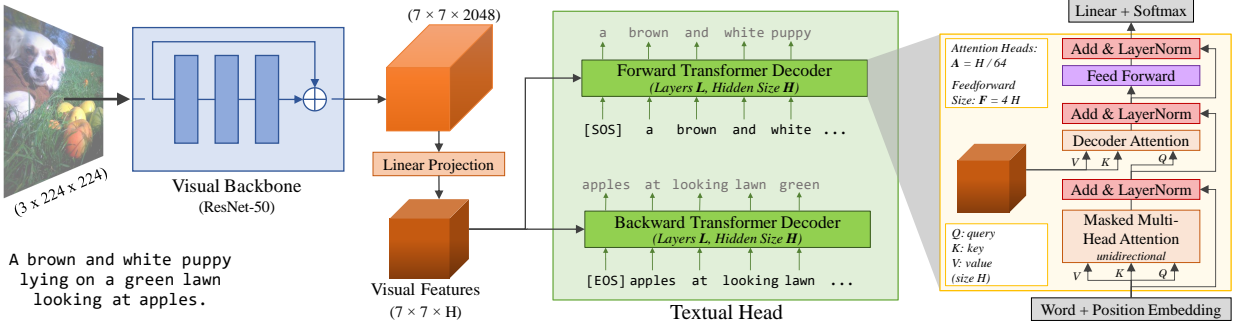


Figure 2.3: **VirTex setup:** Our model consists of a *visual backbone* (ResNet-50), and a *textual head* (two unidirectional Transformers). The visual backbone extracts image features, and textual head predicts captions via bidirectional language modeling (*bicaptioning*). The Transformers perform masked multiheaded self-attention over caption features, and multi-headed attention over image features. Our model is trained end-to-end from scratch. After pre-training, the visual backbone is transferred to downstream visual recognition tasks.

models respectively. After training, we discard the textual head and transfer the visual backbone to downstream visual recognition tasks.

Language Modeling: We choose image captioning [59, 117, 249], as our pre-training task, so far kept *downstream* from vision-based pre-training. We draw inspiration from recent work in NLP using language modeling as a pre-training task to learn transferable text representations. This involves training massive language models – either unidirectional [192] or bidirectional [22, 196, 197, 267], for predicting tokens one by one. However, following BERT [53], many large-scale models [158, 216] instead use *masked language models* (MLMs): some tokens are randomly masked and are predicted by the model. We performed preliminary experiments with MLMs, but like [42, 53] we observed that MLMs converge more slowly than directional models. We note that MLMs have poor sample efficiency, as they only predict a subset of tokens for each caption, while directional models predict all tokens.

Visual Backbone: The visual backbone is a convolutional network that inputs raw image pixels and outputs a grid of image features. During pre-training, these features are used to predict captions. In downstream tasks, we either train linear models on features extracted from the visual backbone or fine-tune the visual backbone end-to-end.

In principle, we could use any convolutional network architecture for the visual backbone. In our experiments, we use a standard ResNet-50 [91] as the visual backbone to facilitate comparison with our baseline methods (Section 2.3). It accepts a 224×224 image and produces a 7×7 grid of 2048-dimensional features after the final convolutional

layer. During pre-training, we apply a linear projection layer to the visual features before passing them to the textual head to facilitate decoder attention over visual features. This projection layer is not used in downstream tasks.

Textual Head: The textual head receives features from the visual backbone and predicts captions for images. It provides a learning signal to the visual backbone during pre-training. Our overall goal is not to predict high-quality captions, but instead to learn transferable visual features.

The textual head comprises two identical language models that predict captions in forward and backward directions respectively. Following recent advances in language modeling, we use Transformers [244], which use multiheaded self-attention both to propagate information along the sequence of caption tokens, as well as to fuse visual and textual features. We closely follow the transformer decoder architecture from [244], but use GELU [101] rather than ReLU, following [53, 196]. We briefly review the architecture here; refer to [244] for a more complete description.

During training, the forward model receives two inputs: image features from the visual backbone, and a caption describing the image. Image features are a matrix of shape $N_I \times D_I$ giving a D_I -dimensional vector for each of the $N_I = 7 \times 7$ positions in the final layer of the visual backbone. As described earlier, the caption $C = (c_0, c_1, \dots, c_T, c_{T+1})$ is a sequence of $T + 2$ tokens, with $c_0 = [\text{SOS}]$ and $c_{T+1} = [\text{EOS}]$. It is trained to predict $C_{1:T+1}$ token-by-token, starting with c_0 . The prediction c_t is *causal* – it only depends on past predictions $c_{0:t-1}$ and visual features. The backward model is similar; it operates right-to-left – trained to predict $C_{T:0}$, given c_{T+1} .

First, we convert the tokens of C to vectors via learned token and positional embeddings, followed by elementwise sum, layer normalization [8] and dropout [224]. Next, we process these vectors through a sequence of Transformer layers. As shown in Figure 2.3, each layer performs masked multiheaded self-attention over token vectors, multiheaded attention between token vectors and image vectors, and applies a two-layer fully-connected network to each vector. These three operations are each followed by dropout, wrapped in a residual connection, and followed by layer normalization. Token vectors interact only through self-attention; the masking in this operation maintains the causal structure of the final predictions. After the last Transformer layer, we apply a linear layer to each vector to predict unnormalized log-probabilities over the token vocabulary.

The forward and backward models consist of independent Transformer layers. However, they share the same token embedding matrix (similar to [192]) which is also reused at the output layers of each model (similar to [111, 194]).

Model Size: Several architectural hyperparameters control the size of our textual head. We can control the *width* of each Transformer layer by varying its *hidden size* H , the number of *attention heads* A used in multiheaded attention, and the *feedforward size* F of the fully-connected network. We follow [53] and always set $A = H/64$ and $F = 4H$; this allows us to control the width of our textual head by varying H . We can also control the *depth* of our textual head by varying the number of transformer layers L .

Tokenization: We lowercase all captions and strip accents from characters, and then tokenize them with SentencePiece [132] using the byte-pair encoding algorithm [213]. The resulting vocabulary comprises 10K tokens, including boundary ([SOS], [EOS]) and out-of-vocab ([UNK]) tokens. Following [196, 197] we restrict subword merges between letters and punctuation to prevent redundant tokens such as ‘dog?’ and ‘dog!’ . Compared to basic tokenization schemes often used for image captioning that split on whitespace [117, 249], BPE makes fewer linguistic assumptions, exploits subword information, and results in fewer out-of-vocab tokens.

Training Details: All models are trained using the train2017 split of the COCO Captions dataset [34], which contains 118K images with five captions each. We apply simple data augmentation during training: randomly crop to 20–100% of the original image size, and apply color jitter (brightness, contrast, saturation, hue). We also apply random horizontal flips, also interchanging the words ‘left’ and ‘right’ in the caption. All images are normalized using the ImageNet color (RGB mean = [0.485, 0.456, 0.406], std = [0.229, 0.224, 0.225]) before passing as input to the visual backbone.

We train using SGD with momentum 0.9 [193, 228] wrapped in LookAhead [281] with $\alpha = 0.5$ and 5 steps. The weight decay is 10^{-4} applied to all parameters except gains and biases in Transformers. We perform distributed training across 8 GPUs with batch normalization [112] per GPU, following [83]. We train with a batch size of 256 images (32 per GPU) for 500K iterations (≈ 1080 epochs). We use linear learning rate warmup [83] for the first 10K iterations followed by cosine decay [161] to zero. We found that the visual backbone required a higher LR than the textual head for faster convergence. The visual backbone uses a max LR of 2×10^{-1} ; the textual head uses 10^{-3} . We implement our models using PyTorch [187] with native automatic mixed-precision [172].

2.3 Experiments

In our experiments, we aim to demonstrate the effectiveness of learning visual representations using language supervision. We train VirTex from scratch on COCO Captions [34] as described in Section 2.2, and evaluate the learned representations on six downstream vision tasks. These evaluations mimic two common styles of transfer learning: linear probe (Section 2.3.1) wherein the visual backbone is kept frozen, and fine-tuning tasks (Section 2.3.2) that allow updating backbone parameters using the downstream dataset.

2.3.1 Linear probe evaluation

We compare VirTex with various pre-training methods to test our two hypotheses: (a) Learning visual features via captions is cheaper than using other types of annotations, like labels and masks. (b) Using semantically dense captions helps with learning effective visual features using fewer training images. We use two standard datasets for linear probe evaluation: PASCAL VOC [67] and ImageNet-1k [208].

PASCAL VOC: We train on VOC07 trainval split (9K images, 20 classes) and report mean average precision (mAP) on test split. Image pre-processing is kept minimal during training and evaluation – we resize the shorter edge to 256 pixels and take a 224×224 center crop. We train per-class SVMs on 2048-dimensional global average pooled features extracted from the last layer of the visual backbone. For training SVMs, we use scikit-learn [190] with LIBLINEAR [68] backend, default parameters are:

```
LinearSVC(  
    cost=C, penalty='l2', dual=True, max_iter=2000, tol=1e-4,  
    class_weight={1: 2, -1: 1}, loss='squared_hinge'  
)
```

We search for the best cost value from $C \in \{0.01, 0.1, 1.0, 10.0\}$ using 2-fold cross-validation on the trainval split. Our evaluation setup is very similar to several works in the self-supervised visual learning literature [28, 83, 175].

ImageNet-1k: Our evaluation protocol resembles MoCo [94], SwAV [28], and many other self-supervised learning works. We train a linear classifier using the ILSVRC 2012 train split and report top-1 accuracy on val split. The classifier is a fully connected layer with softmax, on 2048-dimensional global average pooled features from the visual backbone. The weights are initialized from $N(0.0, 0.01)$, and bias values are initialized to 0.

Method	Annotations	Cost (hours)	PASCAL VOC	ImageNet
MoCo-COCO	self-sup.	–	63.3	41.1
Multi-label Clf.	labels	11.1K [153]	86.2	46.2
Instance Segmentation	masks	30.0K [153]	82.3	51.0
VirTex (1 caption)	captions	1.3K [1]	84.2	50.2
VirTex (5 caption)	captions	6.5K [1]	88.7	53.8

Table 2.1: **Annotation cost efficiency of VirTex:** We compare the downstream performance of various pre-training methods on COCO. Cost estimates are reported in terms of *annotation worker hours* for COCO train2017 split. VirTex outperforms all other methods trained on the same set of images with the best performance vs. cost tradeoff.

We use batch size 256 distributed across 8 GPUs for 100 epochs. We use SGD with momentum 0.9 and weight decay 0, the initial LR to 0.3 and decayed to zero by cosine schedule [161]. For data augmentation during training, we randomly crop 20–100% of the original image size, with a random aspect ratio in $(4/3, 3/4)$, resize to 224×224 , and apply random horizontal flip. During evaluation, we resize the shorter edge to 256 pixels and take a 224×224 center crop.

Evaluation I: Annotation cost efficiency. We compare various pre-training methods using supervision from different types of annotations (Figure 2.2). This allows us to compare the cost efficiency of annotations while using the same set of training images (COCO).

- **MoCo-COCO (self-supervised):** This is a MoCo-v1 model [94] trained on COCO images using the official codebase ¹ and default hyperparameters. Since this model learns only from images, its annotation cost is *zero*.
- **Multi-label Classification (labels):** We use COCO object detection annotations (80 classes), and train a ResNet-50 backbone to predict a K -hot vector with values $1/K$ with a KL-divergence loss, similar to [166]. We estimate the annotation cost of COCO labels from Lin et al. [153] – the *Category Labeling* and *Instance Spotting* steps take $\approx 30K$ hours (328K images). We scale this for 118K images of COCO train2017 split.
- **Instance Segmentation (masks):** We use a pre-trained Mask R-CNN trained from scratch using the COCO dataset [93] from Detectron2 model zoo [256] and extract its ResNet-50 backbone for downstream tasks. Lin et al. [153] mention that collecting 1000 segmentation masks takes 22 worker hours. We scale this estimate for $\approx 860K$ masks in COCO train2017 split, along with label collection cost.
- **VirTex (captions):** We train a VirTex model on COCO Captions, with ResNet-50 visual

¹<https://github.com/facebookresearch/moco>

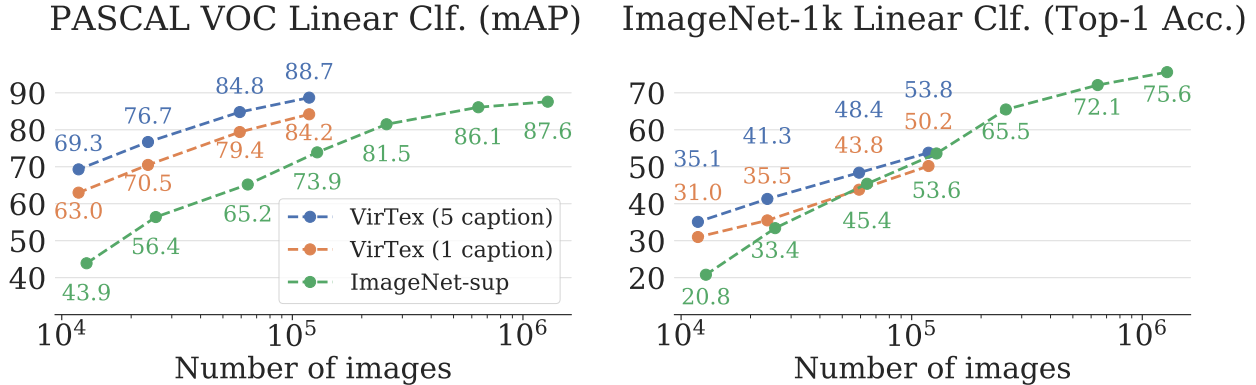


Figure 2.4: **Data efficiency of VirTex:** We compare VirTex and IN-sup models trained using varying amounts of images. VirTex closely matches or significantly outperforms IN-sup on downstream tasks despite using $10\times$ fewer images. ImageNet-supervised models using $\leq 10^5$ images are the mean of 5 trials, std dev. ≤ 1.0 .

backbone and $L = 1, H = 2048$ textual head. Note that COCO Captions provides five captions per image, which effectively increases image-caption pairs by five-fold. Hence for a fair comparison, we also train an additional VirTex model using only one randomly selected caption per image. Cost estimates for COCO Captions are not available in existing literature, to the best of our knowledge. Agrawal et al. [1] report the median time required to write a single caption according to COCO collection protocol as 39.2 seconds. We use this to estimate the cost of collecting $118\text{K} \times 5$ COCO captions.

Results and annotation costs are shown in Section 2.3.1. We observe that VirTex outperforms all methods, and has the best performance vs. cost tradeoff, indicating that learning visual features using captions is more cost-efficient than labels or masks.

Evaluation II: Data efficiency. We believe that the semantic density of captions should allow VirTex to learn effective visual features from fewer images than other methods. To test our hypothesis, we compare VirTex and ImageNet-supervised models (**IN-sup**) trained using varying amounts of images from COCO Captions and ImageNet-1k respectively.

We train 4 VirTex models using random (10%, 20%, 50%, 100%) subsets of COCO (118K images). We also train 4 VirTex models using one randomly selected caption per image. All VirTex models use $L = 1, H = 2048$ textual heads. Training details are the same as before, except that we scale training iterations according to the size of the sampled training set.

As baselines, we train ImageNet-supervised models using randomly sampled subsets of ImageNet (1%, 2%, 5%, 10%, 20%, 50%). Subsets are carefully sampled to mimic the class distribution of full ImageNet. These models are trained following the *exact* training setup used by torchvision models. We use SGD with momentum 0.9 and weight decay 10^{-4} .

Method	Pre-train Images	Annotations	PASCAL VOC	ImageNet
MoCo-IN v1 [94]	1.28M	self-sup.	79.4	60.8
PCL v1 [145]	1.28M	self-sup.	83.1	61.5
SwAV (200 ep.) [28]	1.28M	self-sup.	87.9	72.7
ICMLM (att-fc) [23]	118K	captions	87.5	47.9
VirTex	118K	captions	88.7	53.8

Table 2.2: **System-level comparisons with VirTex:** VirTex is competitive with recent SSL methods and concurrent work while using fewer images with language supervision.

We use a total batch size of 256, and distributed across 8 GPUs. We train for 90 epochs, with an initial learning rate 0.1, which is divided by 10 at epochs 30 and 60.

Results are shown in Figure 2.4. On VOC07, VirTex-100% outperforms IN-sup-100% (mAP **88.7** vs **87.6**), despite using $10\times$ fewer images (118K vs. 1.28M). When using a similar amount of images, VirTex consistently outperforms IN-sup (**blue, orange** vs **green**), indicating superior data efficiency of VirTex. We also observe that given the same number of captions for training, it is better to spread them over more images – VirTex-50% (1 caption) significantly outperforms VirTex-10% (5 captions) (mAP **79.4** vs **69.3**). Comparison with IN-sup on ImageNet-1k classification is unfair for VirTex since IN-sup models are trained for the downstream task, using the downstream dataset. Even so, VirTex-100% outperforms IN-sup-10% (**53.8** vs. **53.6**, 118K vs. 128K images), and consistently outperforms it when both methods use fewer than 100K images.

Evaluation III: ImageNet vs. Cropped COCO. The ImageNet dataset mostly contains *centered* images with a single object, commonly called *iconic* images. On the other hand, the COCO dataset contains ~ 2.9 object classes and ~ 5.7 instances per image. It may seem that VirTex requires fewer images than ImageNet-supervised models as they contain multiple objects per image. We perform a simple comparison to control the varying image statistics between datasets. We crop objects from COCO images and create a dataset of 860K *iconic* images. To closely mimic ImageNet-like images, we randomly expand bounding boxes on all sides by 0–30 pixels before cropping. We train a ResNet-50 with the same hyperparameters as ImageNet-supervised models. It achieves **79.1** VOC07 mAP (vs. **88.7** VirTex). This shows that the data efficiency of VirTex does not *entirely* stem from using images containing multiple objects.

Evaluation IV: System-level comparisons. We compare VirTex directly with different pre-training methods that were developed during the undertaking of this chapter.

- **Self-supervised pre-training:** We choose three methods based on their availability and compatibility with our evaluation setup – MoCo [94], PCL [145], and SwAV [28]. We select officially released model checkpoints that were trained with a similar compute budget as ours and evaluate them with our implemented protocol.
- **ICMLM (Concurrent Work):** We show numbers from *Sariyildiz et al.* [23]; evaluation may slightly differ. This model uses pre-trained BERT [53] for textual features.
- **Vision-language pre-training works:** Since we use captions, we also consider methods that learn multimodal representations for downstream vision-language tasks [36, 143, 147, 149, 164, 225, 231, 288]. As described in Section 2.4, all these methods use an object detector with ImageNet pre-trained backbone. These features are kept frozen, and do not learn from any language supervision at all. Our comparison with ImageNet-supervised models subsumes this family of models.

Results are shown in Table 2.2. VirTex outperforms all methods on VOC07, despite being trained with much fewer images. On ImageNet-1k, the comparison between self-supervised models and VirTex is unfair on both ends, as the former observes downstream images during pre-training, while the latter uses annotated images.

2.3.2 Fine-tuning based evaluation

In this section, we evaluate the visual representations learned by VirTex on tasks that involve fine-tuning the pre-trained model on the downstream task dataset. We consider four tasks with different datasets: (a) Instance Segmentation on COCO [153]; (b) Instance Segmentation on LVIS [86]; (c) Object Detection on PASCAL VOC [67]; and (d) Fine-grained Classification on iNaturalist 2018 [243]. In all these experiments, we use the VirTex model with ResNet-50 visual backbone and a textual head with $L = 1, H = 2048$.

Baselines: Our main baseline in ImageNet-supervised models (IN-sup) similar to linear probe evaluation. We consider three variants of IN-sup pre-trained with $\{10, 50, 100\}\%$ of ImageNet images (Figure 2.4). We also add another baseline – MoCo [95] – a self-supervised visual learning method that, for the first time, showed strong empirical performance over ImageNet-supervised models on fine-tuning based transfer tasks. We include both MoCo-IN (Table 2.2) and MoCo-COCO (Section 2.3.1). Finally, we include a *Random Init* baseline, where the backbone is trained from scratch using the downstream dataset.

We follow the evaluation protocol of MoCo [94] for all four tasks. We use Detectron2 [256] for tasks (a,b,c). Our IN-sup-100% results are slightly better than those reported by He et al. [94] – we use pre-trained ResNet-50 model from torchvision, whereas

Method	Pre-train Images	COCO Instance Segmentation				LVIS Inst Segm		VOC Detection		iNat 18
		AP ^{box}	AP ₅₀ ^{box}	AP ^{mask}	AP ₅₀ ^{mask}	AP ^{mask}	AP ₅₀ ^{mask}	AP ^{box}	AP ₅₀ ^{box}	Top-1
Random Init		36.7	56.7	33.7	53.8	17.4	27.8	33.8	60.2	61.4
IN-sup	1.28M	41.1	62.0	37.2	59.1	22.6	35.1	54.3	81.6	65.2
IN-sup-50%	640K	40.3 _{-0.8}	61.0 _{-1.0}	36.6 _{-0.6}	58.0 _{-1.1}	21.2 _{-1.4}	33.3 _{-1.8}	52.1 _{-2.2}	80.4 _{-1.2}	63.2 _{-2.0}
IN-sup-10%	128K	37.9 _{-3.2}	58.2 _{-3.8}	34.7 _{-2.5}	55.2 _{-3.9}	17.5 _{-5.1}	28.0 _{-7.1}	42.6 _{-11.7}	72.0 _{-9.6}	60.2 _{-4.7}
MoCo-IN	1.28M	40.8 _{-0.3}	61.6 _{-0.4}	36.9 _{-0.3}	58.4 _{-0.7}	22.8 _{+0.2}	35.4 _{+0.3}	56.1 _{+1.8}	81.5 _{-0.1}	63.2 _{-1.7}
MoCo-COCO	118K	38.5 _{-0.6}	58.5 _{-3.5}	35.0 _{-2.2}	55.6 _{-3.5}	20.7 _{-1.9}	32.3 _{-2.8}	47.6 _{-6.7}	75.4 _{-6.2}	60.5 _{-4.4}
VirTex	118K	40.9 _{-0.2}	61.7 _{-0.3}	36.9 _{-0.3}	58.4 _{-0.7}	23.0 _{+0.4}	35.4 _{+0.4}	55.3 _{+1.0}	81.3 _{-0.3}	63.4 _{-1.4}

Table 2.3: **Fine-tuning based evaluation of VirTex:** We compare VirTex with different pre-training methods across four downstream tasks. For each task, all methods use the same architecture. We initialize the ResNet-50 backbone weights from pre-training (except Random Init), which are then fine-tuned end-to-end. Performance gaps with IN-sup are shown on the side. On all tasks, VirTex significantly outperforms all methods that use similar amount of pre-training images. VirTex closely matches or exceeds ImageNet supervised and self-supervised methods, despite using $10\times$ fewer pre-training images.

they used the MSRA ResNet-50 model from Detectron [82]. We briefly describe implementation details that differ from default Detectron2 settings.

COCO Instance Segmentation: We train Mask R-CNN [92] models with ResNet-50-FPN backbones [154]. We initialize the backbone with pre-trained weights, train on train2017 split, and evaluate on val2017 split. We fine-tune all layers end-to-end with BN layers synchronized across GPUs [191] (*SyncBN*). We also use SyncBN in FPN layers. We train with batch size 16 distributed across 8 GPUs, following $2\times$ schedule (180K iterations with initial LR 0.02, multiplied by 0.1 at iterations 120K and 160K).

LVIS Instance Segmentation: The LVIS dataset provides instance segmentation labels for a long tail of 1203 entry-level object categories and stresses the ability to recognize many object types from few training samples. We train Mask R-CNN models with ResNet-50-FPN backbones on train_v1.0 and evaluate on val_v1.0 split. Following MoCo settings, we keep BN parameters frozen for all IN-sup baselines. We train with $2\times$ schedule as COCO, use class resampling and test-time hyperparameters (0.0 score threshold and 300 detections per image) same as [86].

PASCAL VOC Detection: We train Faster R-CNN [204] models with ResNet-50-C4 backbones on trainval07+12 split and evaluate on test2007 split. Like COCO, we fine-tune all models with SyncBN. We train for 24K iterations, including linear LR warmup for the first

100 iterations. We set the maximum LR as 0.02, which is divided by 10 at iterations 18K and 22K. We distribute training across 8 GPUs, with batch size 2 per GPU. We use gradient checkpointing [31, 170] to reduce the heavy memory footprint of these models and train them with the desired batch size on our 12 GB GPUs.

iNaturalist 2018 Fine-grained Classification: The iNaturalist 2018 dataset provides labeled images for 8142 fine-grained categories, with a long-tailed distribution. We fine-tune the pre-trained ResNet-50 with a linear layer end-to-end. Data augmentation and weight initialization are the same as ImageNet-1k linear classification. We train on train2018 split and evaluate on val2018 split, following training setup from torchvision – we train for 100 epochs using SGD with momentum 0.9 and weight decay 10^{-4} , and batch size 256 distributed across 8 GPUs. Fine-tuning uses LR 0.025 (and Random Init uses 0.1), which is multiplied by 0.1 at epochs 70 and 90.

Results: We show results in Table 2.3. VirTex matches or exceeds ImageNet-supervised pre-training and MoCo-IN on all tasks (row 2, 5 vs. 7) despite using $10\times$ fewer pre-training images. Moreover, VirTex significantly outperforms methods that use similar, or more pre-training images (row 3, 4, 6 vs. 7), indicating its superior data efficiency. Among all tasks, VirTex shows significant improvements on IVIS, which shows the effectiveness of natural language annotations in capturing the long tail of visual concepts in the real world.

2.3.3 Ablations

In this section, we conduct ablation studies to isolate the effects of our pre-training setup and modeling decisions and uncover performance trends to seed intuition for future work. We use linear probe evaluation (Section 2.3.1) for all ablation studies.

Ablation I: Pre-training task. We choose bicaptioning task as it gives a dense supervisory signal per caption. To justify this choice, we form three pre-training tasks with *sparser* supervisory signal and compare them with bicaptioning. All variants use ResNet-50 visual backbone and textual heads with transformers having $L = 1, H = 2048$:

- **Forward captioning:** We remove the backward transformer decoder and only perform forward captioning, *i.e.*, predicting caption only in left-to-right direction.
- **Token classification:** We replace the textual head with a linear layer and perform multi-label classification (Section 2.3.1, row 2). We use the set of caption tokens as targets, completely ignoring the linguistic structure of captions.

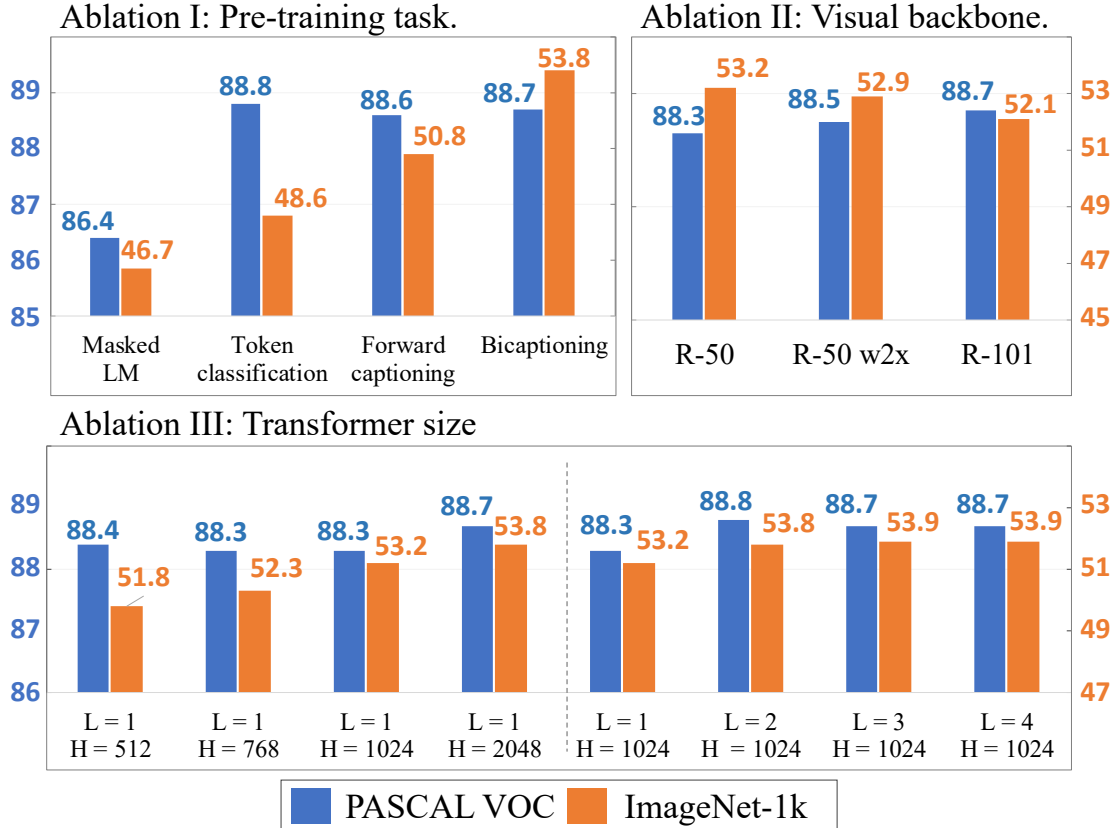


Figure 2.5: **Ablations:** (i) **Pre-training task.** Bicaptioning improves over weaker pre-training tasks – forward captioning, token classification and masked language modeling. (ii) **Visual backbone.** Bigger visual backbones improve downstream performance – both, wider (R-50 w2 \times) and deeper (R-101). (iii) **Transformer size.** Larger transformers (wider and deeper) improve downstream performance.

– **Masked language modeling (MLM):** We use a single bidirectional transformer in the textual head and perform BERT-like masked language modeling. We randomly mask 15% of input tokens, and train the model to predict them.

Results are shown in Figure 2.5(a). Bicaptioning outperforms forward captioning, indicating that denser supervisory signal from bidirectional modeling is beneficial. Bicaptioning and forward captioning both outperform token classification, demonstrating that learning to model the sequential structure of language improves visual features. MLM underperforms all three methods, possibly due to poor sample efficiency, as discussed in Section 2.2.

Ablation II: Visual backbone. Bigger visual backbones tend to show improvements on many vision tasks [91, 92, 261]. We investigate whether they can also benefit VirTex models. We train three VirTex models with visual backbones of varying capacity: (a) ResNet-50 (default), (b) ResNet-50 w2 \times [275] (2 \times channel width), and (c) ResNet-101

($2\times$ depth). All use $L = 1, H = 1024$ textual heads. Results in Figure 2.5(b) show that bigger backbones improve on VOC07 but underperform on ImageNet-1k. We believe it to be an optimization issue – as an additional check, we evaluated on PASCAL VOC object detection (Section 2.3.2) and found that bigger backbones consistently perform better (AP₅₀: ResNet-50 = **81.2**, ResNet-50 w $2\times$ = **82.0**, and ResNet-101 = **82.1**).

Ablation III: Transformer size. Prior work in language modeling has shown that larger Transformers tend to learn better *textual* features [22, 158, 197, 216]. We investigate whether this holds for VirTex: do larger transformers in the textual head cause the visual backbone to learn better *visual* features? As discussed in Section 2.2, we may scale our textual head by increasing its *width* (hidden size H) or its *depth* (number of layers L). We investigate both, training VirTex models with: (a) Fixed $L = 1$, increasing $H \in \{512, 768, 1024, 2048\}$, and (b) Fixed $H = 1024$, increasing $L \in \{1, 2, 3, 4\}$.

Results are shown in Figure 2.5(c) – increasing transformer size, both width and depth, generally improves downstream performance. Performance degrades slightly with very deep transformers ($L = 4$), indicating overfitting. We hope that massive transformers with billions of parameters will help when scaling VirTex to large-scale, more noisy image-text paired datasets [168, 185, 215] that are larger than COCO Captions.

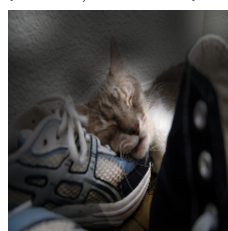
2.3.4 Image captioning

Our goal is to learn transferable visual features via textual supervision. To do so, we use image captioning as a pre-training task. Although our goal is not to advance the state-of-the-art in image captioning, in Figure 2.6 we show quantitative and qualitative results of VirTex models trained from scratch on COCO. All models show modest performance, far from current state-of-the-art methods, that commonly involve some pre-training. However, captioning metrics (CIDEr [245] and SPICE [3]) are known to correlate weakly with human judgment – we surpass human performance on COCO.

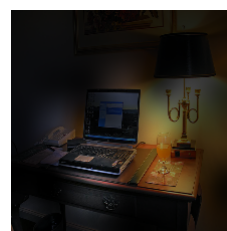
We show some predicted captions by VirTex (R-50, $L = 1, H = 512$) model. We apply *beam search* on the forward transformer decoder (5 beams) to decode the most likely captions. The *decoder attention module* in this transformer attends over a 7×7 grid of image features through $A = 8$ heads at each time-step for predicting a token. We average these 7×7 attention weights over all the heads, and overlay them on 224×224 input image (via bicubic upsampling). Figure 2.6 shows attention visualizations for some tokens. We observe that our model attends to relevant image regions for making predictions, indicating that VirTex learns meaningful visual features with good semantic understanding.

Backbone	Depth	Width	CIDEr	SPICE
R-50	1	512	103.2	19.3
R-50	1	768	103.7	19.6
R-50	1	1024	103.5	19.8
R-50	1	2048	104.2	19.9
R-50	1	1024	103.5	19.8
R-50	2	1024	106.9	20.0
R-50	3	1024	104.3	19.5
R-50	4	1024	103.8	19.2
R-50 w2×	1	1024	102.7	19.6
R-101	1	1024	106.6	20.1

VirTex predicted captions (ResNet-50)
 ($L = 1, H = 512$) forward transformer decoder



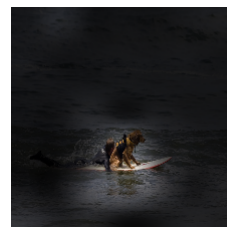
a cat laying on a pair of
 blue shoes



a laptop computer
 sitting on top of a desk



an orange is sitting on
 the side of a road



a dog riding on a
 surfboard in the ocean

Figure 2.6: **Image captioning with VirTex:** We report the image captioning performance of VirTex models on COCO val2017 split, and some model-predicted captions. For the highlighted words, we visualize decoder attention weights from the textual head on the input image. Our model focuses on relevant image regions to predict objects (*shoes*, *desk*), background (*road*) as well as actions (*riding*).

Figures 2.7 and 2.8 contain more examples showing decoder attention weights overlaid on input images.

2.4 Related Work

This chapter relates to several efforts toward moving beyond supervised pre-training on ImageNet, using alternate data sources or pre-training tasks.

Weakly Supervised Learning scales beyond supervised pre-training with a *quantity over quality* approach, and learns on large numbers of images with noisy labels from web services. *Li et al.* [142] trains visual N-gram models on the YFCC-100M dataset [236], that provides 100M Flickr images with user-provided tags. Recent works [124, 227, 260] also use JFT-300M [227] dataset, curated by automatic labeling of images from web signals using Google’s internal tooling. Weakly-supervised learning has also been studied on up to 3.5B Instagram images, using hashtags as labels [166, 265]. These approaches learn visual representations with large quantities of images with low-quality labels; in contrast

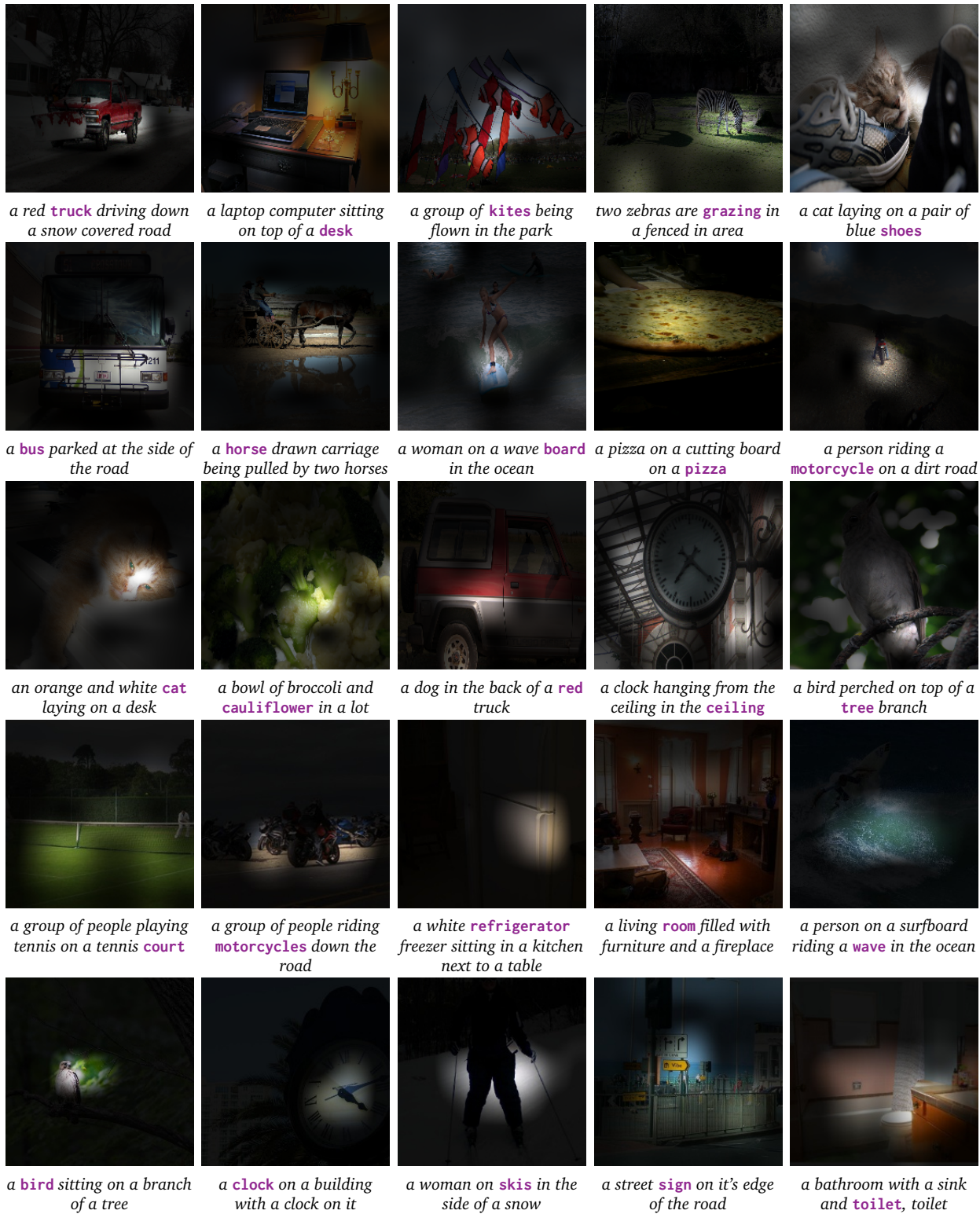


Figure 2.7: We decode captions from the forward transformer of $L = 1, H = 512$ VirTex model using beam search. For the highlighted word, we visualize the decoder attention weights overlaid on the input image.

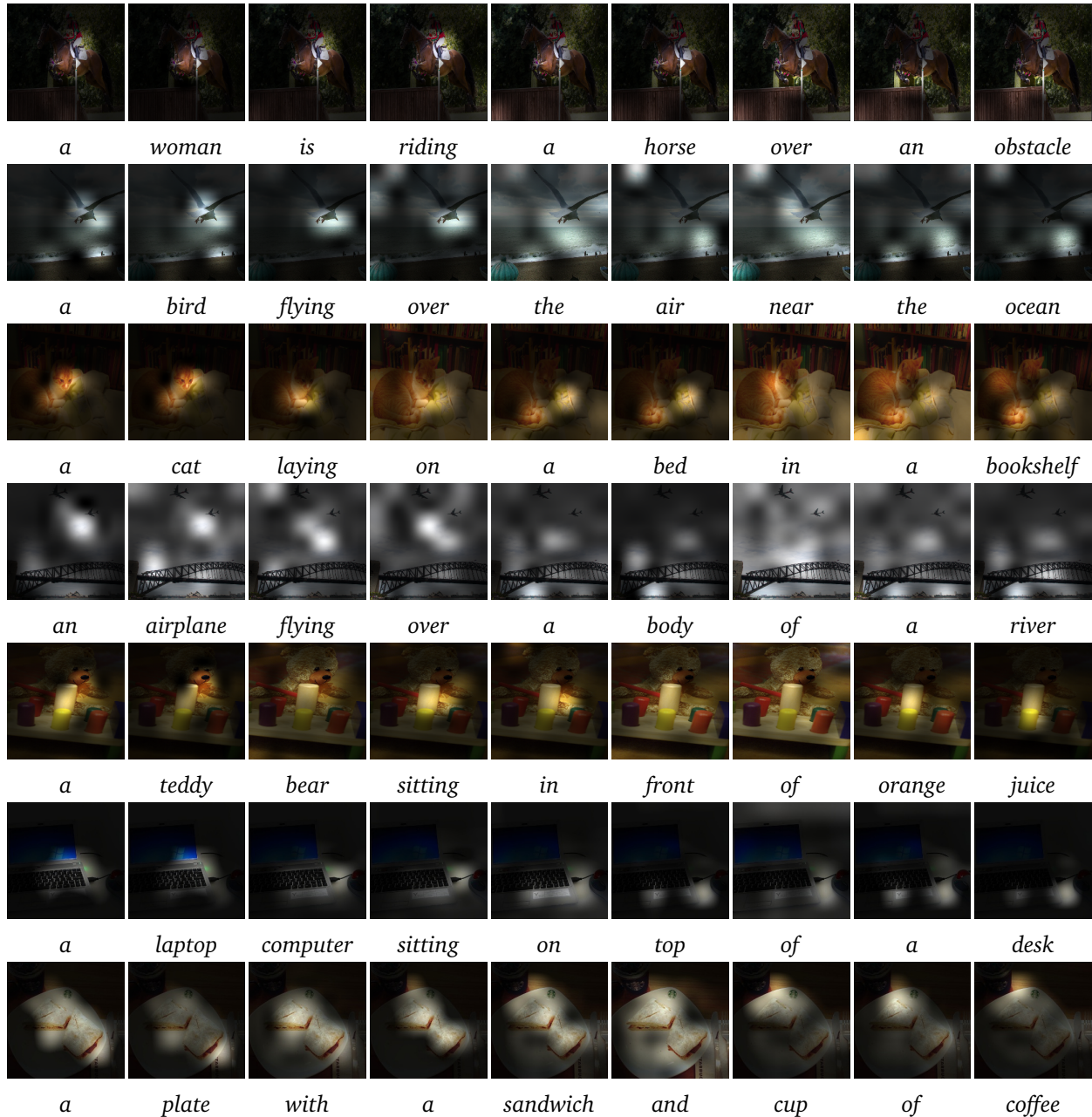


Figure 2.8: Attention visualizations per time step for predicted caption. We decode captions from the forward transformer of $L = 1, H = 512$ VirTex model using beam search. We normalize the attention masks to $[0, 1]$ to improve their contrast for better visibility.

we focus on using fewer images with high-quality annotations.

Self-Supervised Learning focuses on learning visual representations by solving *pretext tasks* defined on unlabeled images. Early works on self-supervised learning proposed hand-crafted pretext tasks, such as context prediction [56], colorization [282, 283], solving jig-

saw puzzles [182], predicting rotation [79], inpainting [188], clustering [26], and generative modeling [57]. Recent works are based on contrastive learning [87, 88], encouraging similarity between image features under different random transformations on single input image [32, 94, 175, 257, 269]. Other approaches use contrastive losses based on context prediction [98, 183], mutual information maximization [9, 103, 237], predicting masked regions [241], and clustering [28, 145, 292].

These methods lack semantic understanding as they rely on low-level visual cues (color, texture), whereas we leverage textual annotations for semantic understanding. Unlike these methods, our approach can leverage additional metadata such as text, when scaled to internet images [168, 185, 215].

Vision-language Pre-training attempts to learn joint representations of image-text paired data that can be transferred to multimodal downstream tasks such as visual question answering [6, 84, 109, 291], visual reasoning [226, 277], referring expressions [119], and language-based image retrieval [271]. Inspired by the success of BERT [53] in NLP, several recent methods use Transformers [244] to learn transferable joint representations of images and text [36, 143, 147, 149, 164, 225, 231, 288].

These methods employ complex pre-training pipelines: they typically **(1)** start from an ImageNet-pre-trained CNN; **(2)** extract region features using an object detector fine-tuned on Visual Genome [128], following [4]; **(3)** optionally start from a pre-trained language model, such as BERT [53]; **(4)** combine the models from (2) and (3), and train a multimodal transformer on Conceptual Captions [215]; **(5)** fine-tune the model from (4) on the downstream task. In this pipeline, all vision-language tasks are downstream from the initial visual representations learned on ImageNet. In contrast, we pre-train via image captioning and put vision tasks downstream from vision-language pre-training.

2.5 Conclusion

In this chapter, we have shown that language-supervised visual learning can be competitive with methods based on supervised classification and self-supervised learning on ImageNet. Training with image-caption pairs opens a clear pathway to scaling our method to orders of magnitude more data from the internet, which we shall discuss in the next chapter.

Chapter 3

Web-curated Image-Text Data from Reddit



Figure 3.1: **RedCaps dataset** comprises 12 million image-text pairs from 350 subreddits. RedCaps data contains everyday things that users like to share on social media, e.g., hobbies (`r/crafts`) and pets (`r/shiba`). Captions often contain specific and fine-grained descriptions (*northern cardinal*, *taj mahal*). Subreddit names provide image labels (`r/shiba`) even when captions may not (*mlem!*), and sometimes group many visually unrelated images through a common semantic meaning (`r/perfectfit`).

3.1 Introduction

Large datasets of image-text pairs from the web have enabled successful transfer learning applications in computer vision. Two such prominent datasets – SBU [185] and Conceptual Captions [215] – are widely used for pre-training vision-and-language representations [36, 108, 143, 147, 149, 164, 225, 231, 288] that transfer to a variety of downstream vision-language tasks like visual question answering [6, 109, 291], visual reasoning [226, 277], and image captioning [1, 34]. Chapter 2 and concurrent work of Bulent Sariyildiz et al. [23] also shows that image-text data from COCO [34] can be used to learn *visual* features that are competitive with supervised pre-training [91] on ImageNet [47, 208] when transferred to downstream tasks [67, 86, 153, 243, 287]. More

recently, CLIP [198] and ALIGN [114] unlock zero-shot image classification ability by scaling up to 400 million and more than a billion image-text pairs respectively.

These datasets have an appealing advantage – they are free from expensive annotations. However, they apply complex filtering steps to deal with noisy web data. For example, Conceptual Captions (CC-3M [215], CC-12M [30]) discard captions without nouns, or whose nouns do not match with image labels predicted by in-house image taggers. They also perform text pre-processing like replacing proper nouns with common nouns. These pipelines are data-inefficient – for example, CC-3M collected 5 billion image-text pairs and filtered them down to 3.3 million. CLIP and ALIGN scale primarily by *relaxing* such filtering, resulting in gargantuan datasets which could be extremely noisy.

How can we obtain high-quality image-text data from the web *without* complex data filtering? We argue that the quality of data depends on its *source* and the *intent* behind its creation. Revisiting data sources, SBU query Flickr with predefined keywords while CC-3M and CC-12M extract images and HTML alt-text from an unspecified set of web pages; CLIP and ALIGN give only vague descriptions of their data sources, and their datasets are non-public. In these sources, text is secondary to images: Flickr focuses on photos, and alt-text is an oft-overlooked *fallback* when images cannot be viewed that frequently contains metadata or generic text (e.g. “alt img” [114]). To obtain higher-quality data, we look for sources where humans use both images and text equally for interaction on the web.

In this chapter, we explore the Reddit [201] social media platform for collecting image-text pairs. Textual data from Reddit is already used for pre-training massive language models [22, 196, 197, 250] in NLP. We collect images and their captions from various topic-specific subreddits. Our dataset of image captions from Reddit (RedCaps in short) consists of 12 million image-text pairs submitted in 350 subreddits between 2008–2020. Figure 3.1 shows some examples from RedCaps – the captions are more conversational, humorous, emotional, and generally more diverse than HTML alt-text.

Apart from linguistic diversity, Reddit offers many other advantages. Subreddits provide additional image labels and group related content – manually selecting subreddits allows us to steer dataset contents without labeling individual instances. Reddit’s *voting* system gives free and organic quality control: unappealing or spam content is actively *downvoted* by users or removed by moderators. RedCaps is one of the largest public image-text datasets, but it is not *static*: we plan to release regular updates with newly uploaded Reddit content, allowing RedCaps to *grow* over time.

We claim that captions written with the intent of human interaction on Reddit are a better source of data than used in other image-text datasets. To this end, we use VirTex (Chapter 2) to learn visual representations by training image captioning models from

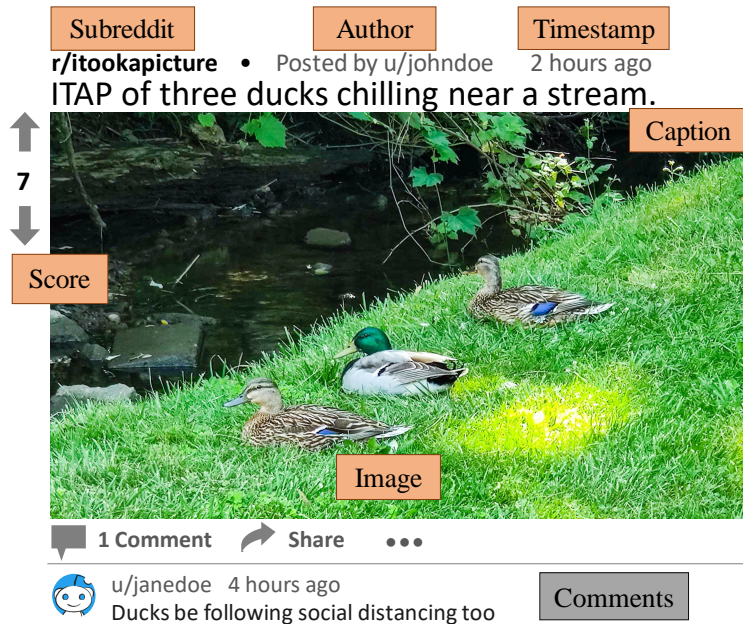


Figure 3.2: **Preview of a Reddit image post:** We collect the RedCaps dataset by downloading images and associated metadata (highlighted in orange) from Reddit image posts.

scratch. We find that human evaluators prefer captioning outputs from models trained on RedCaps vs CC-3M. We also transfer the learned features to **eleven** different downstream datasets for tasks including image classification, object detection, instance segmentation, and fine-grained recognition using both fine-tuning and language-based zero-shot classification [198]. We show that features learned on RedCaps outperform those learned on SBU or CC-3M, demonstrating the utility of our data collection strategy.

3.2 RedCaps: Collecting image-text pairs from Reddit

Reddit is the singular data source for RedCaps. This leads to a very different data collection pipeline than datasets based on HTML alt-text or search engine results.

Overview of Reddit: Reddit is a social media platform for content sharing and discussion. It comprises user-run communities called *subreddits* that cover diverse topics like animals ([r/cats](#), [r/foxes](#)), food ([r/pizza](#), [r/sushi](#)), leisure ([r/hiking](#), [r/crafts](#)), and utility ([r/ceramics](#), [r/tools](#)). Users submit new posts or share existing posts from other subreddits (*cross-posting*), comment and upvote (or downvote) posts to express their interest. We are specifically interested in posts containing images. Figure 3.2 shows an image post submitted by user [u/johndoe](#) in subreddit [r/itookapicture](#). It comprises an image, caption, score (upvotes minus downvotes), and information about the author and time of post creation.

We extract this metadata from millions of image posts to build RedCaps.

Reddit posts also have associated comment threads. These are usually casual conversations *loosely* based on the image. In Figure 3.2, the comment describes ducks as following *social distancing* – it includes context beyond the image (COVID-19 pandemic) and conveys it with a witty remark. Prior works in dialog modeling and text summarization have trained on Reddit comments [2, 55, 99, 171, 250]. For RedCaps, we only use captions as textual data and leave comments for future work.

3.2.1 Data collection pipeline

The uniform structure of subreddits in Reddit allows us to parallelize data collection as independent tasks. Each atomic task involves collecting posts submitted to a single subreddit in one year. Our collection pipeline has three steps: (1) subreddit selection, (2) image post filtering, and (3) caption cleaning.

Step 1. Subreddit selection: We collect images and associated metadata from a manually curated set of subreddits. Subreddits have their own rules, community norms, and moderators so curating subreddits allows us to steer the dataset’s composition without annotating individual instances. We select subreddits with a high volume of image posts, where images tend to be photographs (rather than memes, drawings, screenshots, etc) and post titles tend to describe image content (rather than making jokes, political commentary, etc). We do not select any NSFW, banned, or quarantined subreddits. We want to minimize the number of *people* that appear in RedCaps, so we omit subreddits whose primary purpose is to share or comment on images of people (such as celebrity pics or user selfies). We choose subreddits focused on general photography ([r/pics](#), [r/itookapicture](#)), animals ([r/axolotls](#), [r/birdsofprey](#), [r/dachshund](#)), plants ([r/roses](#), [r/succulents](#)), objects ([r/classiccars](#), [r/trains](#), [r/sneakers](#)), food ([r/steak](#), [r/macarons](#)), scenery ([r/cityporn](#)¹, [r/desertporn](#)), or activities ([r/carpentry](#), [r/kayaking](#)). In total we collect data from 350 subreddits; the full list can be found in Appendix A.1.

Step 2. Image post filtering: We use Pushshift [14] and Reddit [202, 203] APIs to download all image posts submitted to our selected subreddits from 2008–2020. Posts are collected at least six months after their creation to let upvotes stabilize. We only collect posts with images hosted on three domains: Reddit ([i.redd.it](#)), Imgur ([i.imgur.com](#)), and Flickr ([staticflickr.com](#)). Some image posts contain multiple images (*gallery posts*) – in

¹Many subreddits are jokingly include ‘*porn*’ in their name to indicate beautiful non-pornographic images.

this case, we only collect the first image and associate it with the caption. We discard posts with < 2 upvotes to avoid unappealing content, and we discard posts marked NSFW (by their authors or subreddit moderators) to avoid pornographic or disturbing content.

Step 3. Caption cleaning: We expect Reddit post titles to be less noisy than other large-scale sources of image captions such as alt-text [30, 215], so we apply minimal text cleaning. We lowercase captions and use `ftfy` [222] to remove character accents, emojis, and non-latin characters, following [22, 197, 198]. Then we apply simple pattern matching to discard all sub-strings enclosed in brackets (`(.*)`, `[.*]`). These sub-strings usually have non-semantic information: *original content* tags `[oc]`, image resolutions `(800x600 px)`, camera specs `(shot with iPhone)`, self-promotion `[Instagram: @user]`, and other references `(link in comments)`. Finally, like CC-12M [30] we replace social media handles (words starting with `@`) with a `[USR]` token to protect user privacy and reduce redundancy. Due to such filtering, $\approx 12\text{K}$ (0.1%) captions in our dataset are empty strings. We do not discard them, as subreddit names alone provide meaningful supervision. Unlike CC-3M or CC-12M that discard captions without nouns or that don't overlap image tags, we do not discard any instances in this step.

Through this pipeline, we collect 13.4M instances from 350 subreddits. Our collection pipeline is less resource-intensive than existing datasets – we do not require webpage crawlers, search engines, or large databases of indexed webpages. RedCaps is easily extensible in the future by selecting more subreddits and collecting posts from future years. Next, we perform additional filtering to mitigate user privacy risks and harmful stereotypes in RedCaps, resulting in final size of 12M instances.

3.2.2 Ethical considerations

There has been growing awareness about potential biases and harms that can arise from internet-scale image and text datasets [15, 18, 46, 76, 115, 189, 266]. There is a fundamental tension in such datasets: the use of internet data is motivated by the desire to use datasets larger than can be manually annotated or verified, but this also means that such datasets cannot be fully controlled or curated by their creators. We identify two potential risks with RedCaps – privacy of people appearing in RedCaps images, and harmful stereotypes – and attempt to minimize them by *automatic data filtering*. We also discuss the impact of data curation from Reddit on user consent and data distribution in RedCaps.

	Detected (Filtered)	Precision		Missed detections	
		5K	(%)	50K	Full dataset
Images with Faces	1.2M	1615	32%	79	≈19K
Images with NSFW	87K	65	1%	1	≈240
Captions with derogatory phrases	24K	–	–	–	–

Table 3.1: **Automatic filtering:** We use detectors to remove nearly 1.4M instances from RedCaps. We estimate the *precision* of these detectors by reviewing 5K random detected images. After filtering, we review 50K random images (out of 12M) to estimate *missed detections*, which we find to be very low. Caption filtering is deterministic (string matching).

Privacy: The individual who *posts* a given photo on Reddit may not be the person *appearing* in said photo; this can pose privacy risks for people who did not expect to appear in images online [18, 266]. Our first method of mitigation is the manual curation of subreddits which are not focused on describing people (Section 3.2.1). As an additional measure, we use RetinaFace [49] to filter images having any face detection with confidence ≥ 0.9 . Results of this filtering are shown in Table 3.1. The number of detections is high (1.2M), however the precision is low (32%) – most detections are masked faces, statues, and animals. Nevertheless, we remove all of these images to reduce privacy risks while minimizing impact on downstream vision tasks. Estimated number of images with faces in filtered RedCaps is extremely low ($\approx 79K$ out of 12M, or 0.6%).

Harmful Stereotypes: Another concern with Reddit data is that images or language may represent harmful stereotypes about gender, race, or other characteristics of people [15, 18, 189]. We select only non-NSFW subreddits with active moderation for collecting data. This stands in contrast to less curated uses of Reddit data, such as for training large language models. A notable example is GPT-2 [197], whose training data includes at least 63K documents from banned or quarantined subreddits which may contain toxic language [77]. We attempt to further reduce harmful stereotypes in two ways:

- **NSFW images:** We use the InceptionV3 [230] model from [135] to filter images detected as *porn* or *hentai* with confidence ≥ 0.9 . Similar to face filtering, we estimated precision of our filtering and estimated amount of missed detections, shown in Table 3.1. The model detects 87K images with low precision ($\sim 1\%$) – most detections are non-NSFW images with pink and beige hues.
- **Potentially derogatory language:** We filter instances whose captions contain words or phrases from a common blacklist [138]. It is important to note that such coarse filtering might suppress language from marginalized groups reclaiming slurs [15]; however, as

RedCaps is not intended to describe people, we believe this is a pragmatic tradeoff to avoid propagating harmful labels.

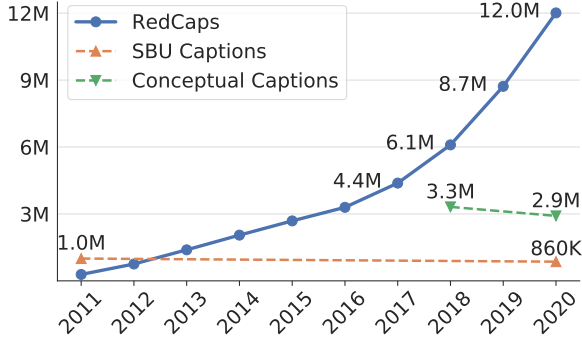
Consent: When submitting to Reddit, users expect their posts to be publicly visible and accessible via the Reddit API we use to download data. However, they did not explicitly consent for their data to be used for training large-scale neural networks [18]. We mitigate this concern in two ways. First, we distribute URLs instead of images; posts deleted from Reddit will thus be automatically removed from RedCaps. Second, we provide a public form allowing anyone to request that specific instances be removed from RedCaps on our website. These decisions mean that over time some images will disappear from RedCaps, making it difficult to *exactly* reproduce experiments in the future. However, we believe this to be less important than allowing users to opt out from RedCaps. Even if images are removed, we expect RedCaps to *grow* over time as we include newer posts (Figure 3.3).

Reddit demographics: Reddit’s user demographics are not representative of the population at large. Compared to US adults, Reddit users skew male (69% vs 49%), young (58% 18-29 years old vs 22%), college educated (36% vs 28%), and politically liberal (41% vs 25%) [13]. Reddit users are predominantly white (63%) [13], and 49% of desktop traffic to Reddit comes from the United States [233]. All of the subreddits in RedCaps use English as their primary language. Taken together, these demographic biases likely also bias the types of objects and places that appear in images on Reddit, and the language used to describe these images. We do not offer explicit countermeasures to these biases, but users of RedCaps should keep in mind that *size doesn’t guarantee diversity* [15].

There may be more subtle issues in our dataset, such as an imbalanced representation of demographic groups [24] or gender bias in object co-occurrence [285] or language [100]. These are hard to control in internet data, so we release RedCaps with explicit instructions on suitable use-cases; specifically requesting models not be trained to identify people, or make decisions that impact people. We document these instructions and other terms-of-use in a datasheet [76], provided in Appendix A.2.

3.3 RedCaps data analysis

Dataset size: Figure 3.3 (top) shows the growth of RedCaps between 2011–2020 based on creation timestamps of image posts (see Figure 3.2). We observe that both SBU and CC-3M have shrunk in size since their release. Since these datasets have released images as URLs (similar to us), an instance would become invalid if the underlying image is removed



Datasets in 2021	# Instances	Released
RedCaps (ours)	12,011,111	✓
CC-12M [30]	12,423,374	✓
WIT-english [223]	5,500,746	✓
CLIP [198]	400M	×
ALIGN [114]	1.8B	×

Figure 3.3: RedCaps was one of the largest public image-text datasets at the time of its creation. Unlike other datasets, it is expected to *grow* over time.

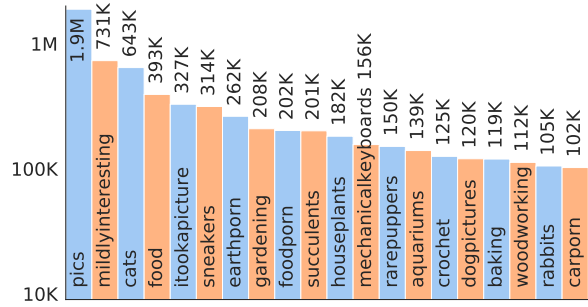


Figure 3.4: Top 20 subreddits with most image-text pairs in RedCaps.

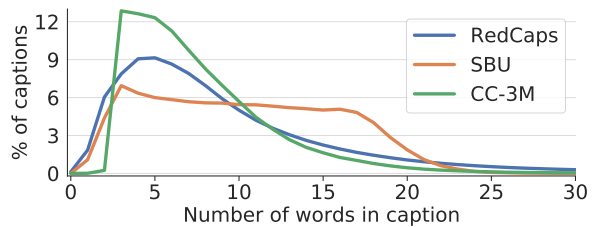


Figure 3.5: RedCaps has a long-tailed distribution of caption lengths.

from the URL ². Likewise, some instances in RedCaps can also disappear in the future if Reddit users delete their posts. However, new image posts on Reddit outnumber deleted posts – we expect RedCaps size to increase in future versions.

Figure 3.3 (bottom), compares RedCaps with recent image-text datasets released in 2021. RedCaps is 2× larger than the English subset of multilingual Wikipedia image-text dataset [223], and nearly as large as CC-12M [30]. Based on current trends, we expect RedCaps to outsize CC-12M by the end of 2021. While CLIP [198] and ALIGN [114] used orders of magnitude larger training datasets, they are not released for public use – RedCaps remains one of the largest public image-text datasets.

Subreddit distribution: RedCaps instances are distributed across 350 subreddits in a long-tail distribution. In Figure 3.4, we show top 20 subreddits with most instances in RedCaps. Subreddit sizes highly correlate with their popularity on Reddit, which depends on what users find interesting to view and share on social media. Large subreddits are based on general photography ([r/pics](#), [r/mildlyinteresting](#), [r/itookapicture](#)), while specific subreddits show that Reddit users enjoy sharing images of food ([r/food](#), [r/foodporn](#)), cute pets ([r/cats](#), [r/dogpictures](#), [r/rabbits](#)), and show off their hobbies ([r/gardening](#),

²We use full SBU and CC-3M annotations for analysis instead of discarding captions with invalid URLs.

Dataset	Unigrams	Bigrams	Trigrams
SBU	28,989	107,847	99,687
CC-3M	21,223	230,077	287,017
RedCaps	95,777	770,100	866,243

Top-5 frequent Trigrams	
SBU	in front of, black and white, in the sky in the background, in the water
CC-3M	a white background, on a white, image may contain, illustration of a may contain person
RedCaps	itap of a, i don't, one of my itap of the, this is my

Table 3.2: Number of $\{1, 2, 3\}$ -grams occurring at least 10 times (**top**) and top-5 trigrams in each dataset (**bottom**).

[r/crochet](#), [r/baking](#)) and accesories ([r/sneakers](#), [r/mechanicalkeyboards](#), [r/carporn](#)). This gives a distribution of visual concepts encountered by humans in daily life without having to predefine an ontology of object classes.

Distribution of caption lengths: Figure 3.5 compares caption lengths between RedCaps and other datasets. We see that RedCaps has the highest mode length at 5 words (vs 3 for CC-3M, SBU) and a heavier tail of long captions ≥ 25 words. SBU has a fairly flat distribution of captions between 3 and 17 words, likely since they only retain captions with at least one preposition and two words in a manually curated term list; RedCaps and CC-3M captions are not filtered in this way and have more peaked distributions reflecting natural language usage.

Word count statistics: Table 3.2 (**top**) compares linguistic diversity between datasets by computing the number of unique unigrams (words), bigrams, and trigrams occurring at least 10 times. This reveals that CC-3M has surprisingly little linguistic diversity, having fewer unique unigrams than SBU despite having $\approx 3\times$ more captions. RedCaps has the most unique terms, with more than $4\times$ unigrams and more than $3\times$ bigrams and trigrams than CC-3M. Greater linguistic diversity means that models trained on RedCaps should recognize a larger variety of visual concepts.

Table 3.2 (**bottom**) shows the most frequent trigrams per dataset. SBU has many prepositional phrases, likely since they require all captions to contain a preposition. Common

Dataset	C. Nouns	P. Nouns	Adjectives	Verbs
SBU	12,985	8,748	2,929	2,497
CC-3M	8,116	654	4,676	3,467
RedCaps	26,060	38,405	11,029	6,019

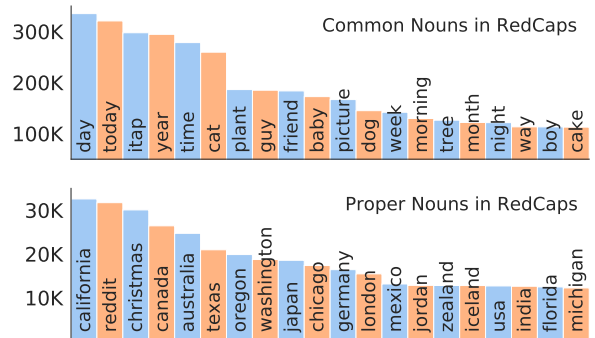


Figure 3.6: Number of unique words by POS, occurring at least 10 times (**top**), and frequent nouns in RedCaps (**bottom**).

CC-3M trigrams *image may contain, may contain person* suggest that the alt-text from which CC-3M takes captions may sometimes be automatically generated. RedCaps trigrams *I don't, one of my, this is my* are more conversational and draw a personal connection between the author and the image, whereas other trigrams *itap of a* and *itap of the* reflect community conventions on [r/itookapicture](#).

Linguistic statistics: We use part-of-speech (POS) tagging to dig deeper into the linguistic diversity of RedCaps. We use the `en_core_web_trf` model from SpaCy [106] to tag POS in all captions. Figure 3.6 (top) shows the number of unique words per POS appearing at least 10 times. RedCaps has $>2\times$ more common nouns and $>4\times$ more proper nouns than SBU, and $>2\times$ more adjectives and $>1.5\times$ more verbs than CC-3M. Nouns in CC-3M are artificially deflated, since their pipeline replaces proper nouns and named entities with hypernyms (which may explain their low unigram counts in Table 3.2).

Figure 3.6 (bottom) shows the most frequent nouns in RedCaps. We see a variety of common nouns including abstract concepts (*day, time*). We find that nouns like *guy, baby,* and *boy* are frequent with RedCaps images with pet animals. Moreover, most frequent proper nouns comprise many cities (*chicago, london*), states (*california, texas*), and countries (*japan, germany, india*), indicating the geographical diversity of RedCaps.

3.4 Experiments

We aim to show that RedCaps offers a unique style of data for both vision and vision-language applications. We demonstrate both applications by adapting VirTex (Chapter 2, [50]), a recent method for pre-training visual representations by performing image captioning as a proxy task. In this section, we measure the effect of data quality on downstream vision tasks by training VirTex models with the same architecture but different datasets – SBU, CC-3M, and RedCaps. To control for RedCaps’s size, we also train on a subset of RedCaps instances from 2020, having comparable size as CC-3M ($3.2M$ vs $2.9M$).

Extending VirTex to VirTex-v2: VirTex comprises an image encoder (*visual backbone*) and a pair of text decoders (*textual head*) that predict the caption token-by-token in forward and backward directions. The base model from [50] used a ResNet-50 [91] visual backbone, and Transformers [244] in textual head that are $L = 1$ layers deep and $H = 2048$ dimensions wide, and was trained on COCO Captions [34] (118K images). We modify this model from [50] to VirTex-v2 to scale to larger noisy datasets, making a few changes described next.

Pre-train Dataset	Pets $N = 37$	Food $N = 101$	Flowers $N = 102$	Cars $N = 196$	Country $N = 211$	SUN $N = 397$	Birdsnap $N = 500$	Average Accuracy
SBU	8.7	3.0	13.7	0.6	0.6	14.7	1.3	6.1
CC-3M	15.5	10.9	10.1	0.5	0.5	33.3	1.6	10.3
RedCaps-20	41.8	54.6	33.5	3.2	2.3	23.9	11.8	24.4
RedCaps	42.4	53.8	26.2	3.1	3.6	26.8	8.3	23.5

Table 3.3: **Zero-shot image classification with VirTex-v2.** We train models of exactly the same capacity using four different image-text datasets, then transfer them zero-shot to **seven** image classification datasets ($N = \#classes$).

- **Model architecture:** We use deeper Transformers with $L = 6$ layers. To balance the memory requirements, we reduce the width to $H = 512$. We use the recent *Pre-LN* Transformer variant [10, 197, 252] that is more stable to train large transformers [262] – LayerNorm [8] is moved inside the residual connection, and we add LayerNorm before the prediction layer.
- **Tokenization:** Similar to VirTex, we use SentencePiece tokenizer [132] with BPE [213]. We build a vocabulary of 30K tokens from the combined caption corpus of SBU, CC-3M and RedCaps. For fair comparison, we use the same vocabulary for all models trained on different datasets. When training with RedCaps, we *prefix* the caption with subreddit tokens: e.g. for Figure 3.1 (**r/birdpics**), the caption becomes ‘[SOS] bird pics [SEP] northern male cardinal [EOS]’. We use wordsegment [113] to break subreddit names to words (e.g. itookapicture \rightarrow i took a picture).
- **Training details:** We use AdamW [122, 162] with weight decay 10^{-2} and max learning rate 5×10^{-4} with a linear warmup for the first 10K iterations, followed by cosine decay [161] to zero. We also use label smoothing ($\epsilon_{ls} = 0.1$) [230] which has improved language generation for machine translation [244]. We train for 1.5M iterations with a total batch size 256 across $8 \times 2080Ti$ GPUs.

We average the last five checkpoints (saved every 2000 iterations) to use for downstream tasks and image captioning. All other details remain unchanged from Chapter 2.

3.4.1 Transfer learning on downstream vision tasks

We evaluate the quality of visual representations learned from SBU, CC-3M, and RedCaps by training VirTex-v2 models on each, then transferring the visual backbone to image classification and instance segmentation on **eleven** different downstream datasets. Our evaluation setup closely follows recent works on self-supervised learning [28, 32, 94] and language-supervised [50, 198] learning.

Zero-shot image classification: Training with language supervision enables *zero-shot* transfer to downstream tasks without *any* additional task-specific training [142, 198]. We evaluate the utility of different datasets for representation learning by comparing zero-shot performance on seven classification datasets: Oxford-IIIT Pets [186], Food-101 [21], Flowers-102 [181], Stanford Cars [126], Country-211 [198], and SUN-397 [258], and Birdsnap [17]. Inspired by CLIP [198], we perform zero-shot classification by designing one *prompt* per category in the target dataset and ranking the log-probabilities predicted for each prompt, averaging predictions from the forward and backward Transformers. For SBU and CC-3M we follow CLIP and use the prompt ‘[SOS] a photo of a/an [label] [EOS]’; for RedCaps we adjust to the training setup and use a prompt with prefixed subreddit – ‘[SOS] i took a picture [SEP] itap of a/an [label] [EOS]’.

Results are in Table 3.3. VirTex-v2 models trained on RedCaps outperform those trained on SBU and CC-3M on **six** out of seven datasets. This is not due to RedCaps’s larger size: models trained on RedCaps-20 also outperform those trained on CC-3M.

Linear probe evaluation: We also evaluate models for image classification on these datasets using linear models trained over *frozen* visual features. Our evaluation details exactly follow CLIP – we use scikit-learn [190] logistic regression with L-BFGS and train for a maximum of 1000 iterations. For each dataset, we hold out a randomly sampled 10% subset of the training data and use it for validation. Similar to CLIP, we start with sweeping L2 regularization parameter $\lambda \in \{10^{-6}, 10^{-4}, 10^{-2}, 1, 10^2, 10^4, 10^6\}$ and select two λ values with the highest top-1 accuracy on held-out split (these were always consecutive in our experiments). We *zoom in* the range with eight equally spaced λ per decade in logarithmic space to find the best value. Finally, we use this λ to train on the combined training data (including held-out 10%) and report top-1 accuracy on the test split. The number of instances in training and test splits used is the same as used for evaluating CLIP. Results are shown in Table 3.4 with similar trends as zero-shot transfer.

Comparison with CLIP: Despite improvements over SBU and CC-3M, our absolute zero-shot performance falls behind CLIP (e.g Food-101 top-1 with ResNet-50 – 81.1 vs. 54.6). Their results are not comparable, as CLIP uses a different architecture (contrastive vs autoregressive), deeper transformer (12 vs 6 layers), larger dataset (400M vs 12M instances), longer training (12.8B image updates vs 384M), and prompt ensembling. Our goal is not to achieve state-of-the-art performance, but instead to compare the impact of different data sources on the quality of learned visual features.

Pre-train Dataset	Pets	Food	Flowers	Cars	Country	SUN	Birdsnap	Average Accuracy
	$N = 37$	$N = 101$	$N = 102$	$N = 196$	$N = 211$	$N = 397$	$N = 500$	
SBU	61.8	48.5	80.3	22.2	12.0	61.3	18.6	43.5
CC-3M	69.9	57.3	76.6	25.2	12.8	70.0	16.1	46.8
RedCaps-20	87.0	79.1	85.9	39.1	11.6	63.6	30.6	56.7
RedCaps	85.0	80.8	86.3	43.9	13.6	67.3	28.1	57.9

Table 3.4: **Linear probe evaluation with VirTex-v2.** We train logistic regression classifiers for **seven** image classification datasets, using frozen visual features extracted from models trained using four different image-text datasets.

Pre-train Dataset	ImageNet Top-1			VOC	COCO	LVIS
	Zero shot	Linear Cls.	k-NN (k=20)	Cls. mAP	Segm. AP	Segm. AP
SBU	5.2	45.5	38.7	85.0	36.5	22.0
CC-3M	20.7	53.9	45.4	87.0	37.2	22.9
RedCaps	22.7	53.4	52.0	87.5	37.0	23.0

Table 3.5: **Additional tasks:** RedCaps trained model matches or exceeds models trained on SBU/CC-3M.

Other tasks: We evaluate on standard transfer tasks with four other datasets: PASCAL VOC and ImageNet-1k linear classification with *frozen* features and instance segmentation [92] on COCO [153] and LVIS [86] with *end-to-end fine-tuning* of Mask R-CNN. These tasks follow the same setup as [50]. On ImageNet, we also perform k nearest neighbor classification ($k=20$), following [29, 257], and zero-shot classification as described above. Results are shown in Section 3.4.1. All models perform similarly on fine-tuning tasks (COCO and LVIS), while RedCaps trained model gains on tasks involving minimal or no fine-tuning – k-NN (52.0 vs 45.4) and zero-shot (22.7 vs 20.7) on ImageNet, and linear classification on VOC (87.5 vs 87.0).

3.4.2 Image captioning

We hope that the unique conversational flavor of RedCaps can enable more human-like and *conversational* image captioning models. We use VirTex-v2 pre-trained models for image captioning – we use nucleus sampling [105] with nucleus size 0.9 to decode a caption from the forward Transformer. In this section, we demonstrate all results on an additional *held-out test set* of 1K instances sampled randomly from image posts submitted to our selected subreddits in the first week of 2021.

Evaluating caption predictions: Standard metrics for automatic image caption evaluation correlate poorly with human judgment [3, 245]. We thus evaluate caption predictions via user studies. We sample captions from models trained on RedCaps and CC-3M, then present crowd workers with the image and both captions. Workers are told that one caption is written by a human and the other machine-generated, and asked to guess which is human-written. We take a majority vote among three workers for each of our 1K test images. Workers preferred captions from the RedCaps-trained model for 633 out of 1000 images. We run a similar study to compare against ground-truth captions, and workers still prefer generated captions for 416 out of 1000 images. See Figure 3.7 for some examples.

Subreddit-conditioned captioning: Captions from different subreddits have distinct styles, focusing on different image aspects or using community-specific jargon. For example, captions in `r/itookapicture` usually start with *itap of ...*. We use this observation to generate captions with distinct styles by prompting a RedCaps-trained model with *different* subreddits. Figure 3.8 shows examples of such diverse captions for images.

3.5 Related work

RedCaps is directly related to recent efforts on building large image-text datasets from the internet without expensive human annotation. Two notable datasets are SBU [185] and Conceptual Captions [215]. Originally intended for image-text retrieval and image captioning, they are now widely used for training generic vision-language representations [36, 108, 121, 143, 147, 149, 164, 225, 231, 288] that transfer to downstream tasks like visual question answering [6, 109, 291], referring expressions [119], and visual reasoning [226, 277]. More recent works build larger datasets specifically for vision-language pre-training, e.g. LAIT [195], Conceptual-12M [30], and Wikipedia-ImageText [223]. Similar to these datasets, RedCaps offers rich semantic data for pre-training applications. However, our choice of data source and hence the data quality is unique.

Image-text datasets are used for visual representation learning. Li et al. [142] trained visual N-gram models on YFCC-100M [236]; VirTex (Chapter 2, [50]) and ICMLM [23] learn features from COCO Captions [34] that are competitive with supervised ImageNet training [91, 131] on many downstream tasks [67, 86, 153, 208, 243], and [114, 198] scale up to very larger non-public datasets that are larger than RedCaps.

A core motivation for collecting image-text data is scaling to larger datasets without bearing annotation costs. Related to this goal are efforts that learn from large quantities

of noisy non-text labels for web images such as WebVision [148], YFCC-100M [236], JFT-300M [40, 102], and Instagram-3.5B [166].

3.6 Conclusion

This chapter has introduced RedCaps, a large-scale dataset of images and captions collected from Reddit. As a source of data, Reddit is appealing: text and images are both created and shared by people, for the explicit purpose of starting a discussion with other people, leading to natural and varied content. Its subreddit structure allows us to manually curate our dataset’s content without labeling individual instances. We utilize this structure to collect a dataset focused on animals, objects, scenery, and activities. We have shown that RedCaps is useful for learning visual representations that transfer to many downstream tasks, including zero-shot settings that use no task-specific training data. We have also shown that RedCaps can be used to learn image captioning models that generate high-quality text of multiple styles.

RedCaps is not without flaws. We have tried to minimize problematic content through subreddit curation and automated filtering, but the unfathomable nature of large data means that RedCaps may contain a small number of instances with NSFW images or harmful language. Reddit’s demographic biases mean that RedCaps may not equally represent all groups. Users should carefully consider these limitations for any new tasks developed on RedCaps. Moreover, users should be especially wary of applications that make predictions about people. Despite these limitations, we hope that RedCaps will help enable a wide variety of new applications and advances in vision and language.

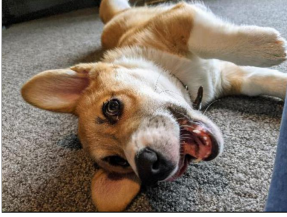

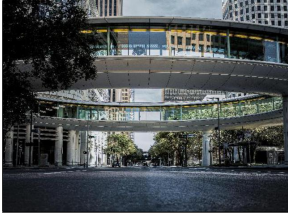


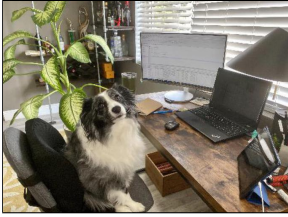



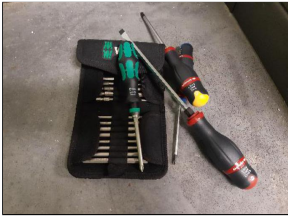





			
CC-3M <i>animal lying on the ground</i>	CC-3M <i>a car is completely covered in snow.</i>	CC-3M <i>the building is a-story polished concrete floor.</i>	CC-3M <i>how to cook a rack of ribs</i>
RedCaps <i>r/lookatmydog: <u>my little guy</u></i>	RedCaps <i>r/mildlyinteresting: <u>this snow sculpture</u></i>	RedCaps <i>r/pics: <u>a building in singapore</u></i>	RedCaps <i>r/foodporn: <u>homemade pizza</u></i>
			
CC-3M <i>the road leading to the mountains</i>	CC-3M <i>hibiscus flower in the dark</i>	CC-3M <i>person , the dog , at the office</i>	CC-3M <i>biological variety uncertain future produce slalom</i>
RedCaps <i>r/mildlyinteresting: <u>this bridge in japan</u></i>	RedCaps <i>r/pics: <u>i took this picture of a hibiscus flower at night</u></i>	RedCaps <i>r/mechanicalkeyboards: <u>my dog is helping me work from home</u></i>	RedCaps <i>r/tea: <u>my first time making matcha green tea!</u></i>
			
CC-3M <i>person - a gray bird sitting on a branch</i>	CC-3M <i>alternative images of this product</i>	CC-3M <i>the wires are now mounted on the wall.</i>	CC-3M <i>a beautiful white water fountain in the mist</i>
RedCaps <i>r/itookapicture: <u>itap of some pigeons</u></i>	RedCaps <i>r/sneakers: <u>thoughts on these?</u></i>	RedCaps <i>r/diy: <u>diy tool bag</u></i>	RedCaps <i>r/pics: <u>my first time seeing snow</u></i>
			
CC-3M <i>this ticket is not only for sale.</i>	CC-3M <i>this is what cats look like.</i>	CC-3M <i>the tallest building complex , is currently under construction .</i>	CC-3M <i><u>this is a beautiful green cactus plant.</u></i>
RedCaps <i>r/mechanicalkeyboards: <u>i'm not sure if i'm doing this left</u></i>	RedCaps <i>r/cats: <u>my cat is helping me study</u></i>	RedCaps <i>r/pics: <u>golden gate bridge</u></i>	RedCaps <i>r/succulents: <u>what is this?</u></i>

Figure 3.7: Image captioning with VirTex-v2 trained on CC-3M vs RedCaps. Three crowd workers have observed these captions (without subreddit names) and voted the caption which seems more likely to be written by a human. The captions voted by majority of workers are underlined. Most of the voted captions are predicted by the RedCaps-trained model. These captions mention (top row): organic references (*little guy* vs *animal*), witty remarks (*snow sculpture*), and specific mentions (*singapore*).








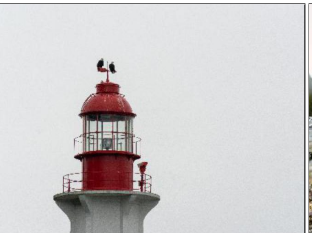

			
r/itookapicture: <i>itap of my dog.</i>	r/itookapicture: <i>itap of my coffee</i>	r/earthporn: <i>sunset in venice, italy</i>	r/earthporn: <i>saturn's north pole.</i>
r/absoluteunits: <i>this absolute unit of a pug</i>	r/absoluteunits: <i>this absolute unit of a coffee.</i>	r/food: <i>a cold beer on the beach.</i>	r/food: <i>the clearest image of saturn</i>
r/somethingimade: <i>i made a bed for my pug.</i>	r/somethingimade: <i>i made a heart latte.</i>	r/pics: <i>shot from a beach house!</i>	r/pics: <i>the clearest image of saturn ever taken</i>
			
r/food: <i>english breakfast</i>	r/food: <i>i'm not sure if these two are getting ready for dinner tonight.</i>	r/food: <i>i made a plant stand for my wife's birthday present</i>	r/food: <i>christmas cat</i>
r/thriftstorehauls: <i>i found this plate at goodwill for \$5</i>	r/thriftstorehauls: <i>found these two pugs in my local thrift store. they are both lonesome and they are so cute.</i>	r/thriftstorehauls: <i>found this beauty for \$20</i>	r/thriftstorehauls: <i>i found a little elf hat for my cat!</i>
r/dogpictures: <i>my dog ate his breakfast today</i>	r/dogpictures: <i>my two pugs snuggling under the couch</i>	r/dogpictures: <i>my dog thinks he's a human</i>	r/dogpictures: <i>merry christmas from my cat</i>
			
r/amateurphotography: <i>i was told you guys would appreciate this.</i>	r/amateurphotography: <i>i took this picture of a highway interchange in china</i>	r/amateurphotography: <i>lighthouse</i>	r/amateurphotography: <i>a waterfall in the rockies</i>
r/vintage: <i>found this guy in my parents garage. he's been sitting in there for years.</i>	r/vintage: <i>i've been looking for a few years now. i finally found a bridge in taiwan.</i>	r/vintage: <i>vintage 2!</i>	r/vintage: <i>my favorite waterfall</i>
r/pics: <i>my owl has been in the same spot since i've been working on my phone.</i>	r/pics: <i>highway interchange between shelbyville and la</i>	r/pics: <i>lighthouse in the fog</i>	r/pics: <i>a waterfall in the rockies</i>

Figure 3.8: **Subreddit-controlled caption style.** We prompt the VirTex-v2 model trained on RedCaps with subreddit names while decoding captions. We observe that such conditioning captures subtle linguistic structures (**r/itookapicture:** *itap of ...*, **r/somethingimade:** *i made...*). or changes the main subject of caption (**r/earthporn:** *venice*, **r/food:** *cold beer*). However, for completely unrelated images (saturn), the model tends to ignore the conditioning while generating captions.

Chapter 4

Hyperbolic Image-Text Representations



Figure 4.1: **Hyperbolic image-text representations.** **Left:** Images and text depict *concepts* and can be jointly viewed in a *visual-semantic hierarchy*, wherein text ‘*exhausted doggo*’ is more generic than an image (which might have more details like a cat or snow). Our method MERU embeds images and text in a hyperbolic space that is well-suited to embed tree-like data. **Right:** Representation manifolds of CLIP (*hypersphere*) and MERU (*hyperboloid*) illustrated in 3D. MERU assumes the origin to represent the *most generic concept*, and embeds text closer to the origin than images.

4.1 Introduction

It is commonly said that ‘*an image is worth a thousand words*’ – consequently, images contain a lot more information than the sentences which typically describe them. For example, given the middle image in Figure 4.1 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 [247], namely, ‘*exhausted doggo*’ → ‘*a cat and a dog playing in the street*’ → (Figure 4.1 middle image). Providing multimodal models access to this inductive bias about vision and language has the potential to improve generalization [198], interpretability [212] and enable better exploratory data analysis of large-scale datasets [198, 211].

Vision-language representation learning has catalyzed a lot of recent progress in computer vision. Methods like CLIP [198] and ALIGN [114] have shown that Transformer-based [244] 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. Such a geometry can find it hard to capture the visual-semantic hierarchy.

An affine Euclidean space treats all embedded points in the same manner, with the same distance metric being applied to all points [177]. 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, 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 [141], making them a continuous relaxation of trees. This allows a generic concept (‘*cat*’) to have many neighbors by placing it close to the origin [179], and more specific concepts further away. Thus, distinct specific concepts like images in Figure 4.1 can be far away from each other while being close to some generic concept (‘*animal*’).

In this chapter, we introduce the first large-scale contrastive image-text models that yield hyperbolic representations [179] – MERU¹ that captures the visual-semantic hierarchy. Our method conceptually resembles current state-of-the-art contrastive methods [114, 198]. Importantly the hierarchy *emerges* in the representation space, given access only to image-text pairs during training.

Practically, MERU confers multiple benefits such as (a) better performance on image

¹Meru 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* (wikipedia.org/wiki/Mount_Meru and wikipedia.org/wiki/Semeru)

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. In summary, this chapter comprises a series of contributions as follows:

- We introduce MERU, the first implementation of deep hyperbolic representations we are aware of, training ViTs [60] with 12M image-text pairs.
- We provide a strong CLIP baseline that outperforms previous re-implementations [176] 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 [134].
- We perform thorough qualitative analysis with MERU to demonstrate its potential for exploratory data analysis of large-scale multimodal datasets.

Our code and models are publicly available at github.com/facebookresearch/meru.

4.2 Preliminaries

We briefly review Riemannian manifolds (Section 4.2.1) and essential concepts of hyperbolic geometry (Section 4.2.2). For a more thorough treatment of the topic, we refer the reader to textbooks by Ratcliffe [200] and Lee [141].

4.2.1 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).

4.2.2 Lorentz model of hyperbolic geometry

We use the Lorentz model of hyperbolic geometry for developing MERU. This model represents a hyperbolic space of n dimensions on the upper half of a two-sheeted hyperboloid in \mathbb{R}^{n+1} . See Figure 4.1 for an illustration of \mathcal{L}^2 in \mathbb{R}^3 . Hyperbolic geometry has a direct connection to the study of special relativity theory [62, 63]. 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 as *space dimensions* [174]. Every vector $\mathbf{x} \in \mathbb{R}^{n+1}$ can be written as $[\mathbf{x}_{space}, x_{time}]$, where $\mathbf{x}_{space} \in \mathbb{R}^n$ and $x_{time} \in \mathbb{R}$.

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

$$\langle \mathbf{x}, \mathbf{y} \rangle_{\mathcal{L}} = \langle \mathbf{x}_{space}, \mathbf{y}_{space} \rangle - x_{time} y_{time} \quad (4.1)$$

The induced *Lorentzian norm* is $\|\mathbf{x}\|_{\mathcal{L}} = \sqrt{|\langle \mathbf{x}, \mathbf{x} \rangle_{\mathcal{L}}|}$. The Lorentz model possessing a constant curvature $-c$ is defined as a following set of vectors:

$$\mathcal{L}^n = \{\mathbf{x} \in \mathbb{R}^{n+1} : \langle \mathbf{x}, \mathbf{x} \rangle_{\mathcal{L}} = -1/c\}, \quad c > 0 \quad (4.2)$$

All vectors in this set satisfy the following constraint:

$$x_{time} = \sqrt{1/c + \|\mathbf{x}_{space}\|^2} \quad (4.3)$$

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 defined as follows:

$$d_{\mathcal{L}}(\mathbf{x}, \mathbf{y}) = \sqrt{1/c} \cdot \cosh^{-1}(-c \langle \mathbf{x}, \mathbf{y} \rangle_{\mathcal{L}}) \quad (4.4)$$

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:

$$\mathcal{T}_{\mathbf{z}}\mathcal{L}^n = \{\mathbf{v} \in \mathbb{R}^{n+1} : \langle \mathbf{z}, \mathbf{v} \rangle_{\mathcal{L}} = 0\} \quad (4.5)$$

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

$$\mathbf{v} = \text{proj}_{\mathbf{z}}(\mathbf{u}) = \mathbf{u} + c \mathbf{z} \langle \mathbf{z}, \mathbf{u} \rangle_{\mathcal{L}} \quad (4.6)$$

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 $\text{expm}_{\mathbf{z}} : \mathcal{T}_{\mathbf{z}}\mathcal{L}^n \rightarrow \mathcal{L}^n$ with the expression:

$$\mathbf{x} = \text{expm}_{\mathbf{z}}(\mathbf{v}) = \cosh(\sqrt{c} \|\mathbf{v}\|_{\mathcal{L}}) \mathbf{z} + \frac{\sinh(\sqrt{c} \|\mathbf{v}\|_{\mathcal{L}})}{\sqrt{c} \|\mathbf{v}\|_{\mathcal{L}}} \mathbf{v} \quad (4.7)$$

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

$$\mathbf{v} = \text{logm}_{\mathbf{z}}(\mathbf{x}) = \frac{\cosh^{-1}(-c \langle \mathbf{z}, \mathbf{x} \rangle_{\mathcal{L}})}{\sqrt{(c \langle \mathbf{z}, \mathbf{x} \rangle_{\mathcal{L}})^2 - 1}} \text{proj}_{\mathbf{z}}(\mathbf{x}) \quad (4.8)$$

For our approach, we will only consider exponential and logarithmic maps for the origin of the hyperboloid, $\mathbf{z} = \mathbf{O} = [\mathbf{0}, \sqrt{1/c}]$.

4.3 Approach

In this section, we discuss the modeling pipeline and learning objectives of MERU to learn hyperbolic representations of images and text. We use the tools of hyperbolic geometry introduced in Section 4.2 throughout our discussion.

Our model design is inspired by a family of contrastive vision-language models like CLIP [198] due to their simplicity and scalability. As shown in Figure 4.2, we process images and text using two separate encoders, and obtain embedding vectors of a fixed dimension n . Beyond this, there are two crucial design choices: (1) transferring embeddings from Euclidean space to the Lorentz hyperboloid, and (2) designing suitable training objectives that induce semantics and structure in the representation space.

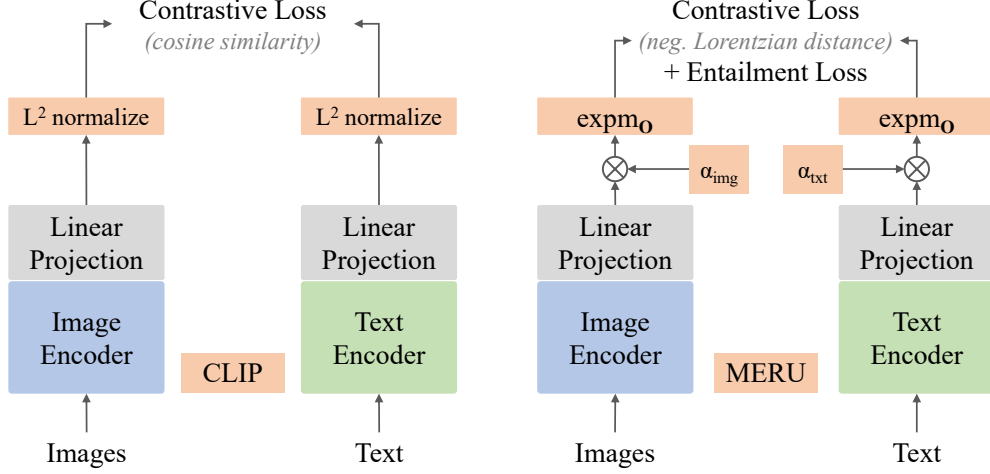


Figure 4.2: **MERU model design:** MERU comprises similar architectural components as standard image-text contrastive models like CLIP. While CLIP projects the embeddings to a unit hypersphere, MERU lifts them onto the Lorentz hyperboloid using the exponential map. The contrastive loss uses the negative of Lorentzian distance as a similarity metric, and an entailment loss enforces ‘*text entails image*’ partial order in the representation space.

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 $\mathbf{v} = [\mathbf{v}_{enc}, 0] \in \mathbb{R}^{n+1}$. We observe that \mathbf{v} belongs to the tangent space at the hyperboloid origin \mathbf{O} , as Eqn. 4.5 is satisfied: $\langle \mathbf{O}, \mathbf{v} \rangle_{\mathcal{L}} = 0$. Thus, we parameterize *only* the *space* components of the Lorentz model. Due to such parameterization, we can simplify the exponential map from Eqn. 4.7 by writing only *space* components:

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

The first term reduces to \mathbf{O} . Moreover, the Lorentzian norm of \mathbf{v} simplifies to the Euclidean norm of *space* components: $\|\mathbf{v}\|_{\mathcal{L}}^2 = \langle \mathbf{v}, \mathbf{v} \rangle_{\mathcal{L}} = \langle \mathbf{v}_{space}, \mathbf{v}_{space} \rangle - 0 = \|\mathbf{v}_{space}\|^2$. This substitution simplifies the above equation as follows:

$$\mathbf{x}_{space} = \frac{\sinh(\sqrt{c} \|\mathbf{v}_{space}\|)}{\sqrt{c} \|\mathbf{v}_{space}\|} \mathbf{v}_{space} \quad (4.9)$$

The corresponding *time* component x_{time} can be computed from \mathbf{x}_{space} using Eqn. 4.3, the resulting \mathbf{x} *always* lies on the hyperboloid. This eliminates the need for an orthogonal projection (Eqn. 4.6) and simplifies the exponential map. Our parameterization is simpler than previous work which parameterizes in full ambient space \mathbb{R}^{n+1} [137, 139, 180].

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 expm_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 (Figure 4.1) 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. [139], that works for arbitrary hyperboloid curvatures $-c$.

4.3.1 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. 4.4) 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 [220] used in CLIP [198] with the crucial difference being that we compute distances on the hyperboloid instead of cosine similarity.

4.3.2 Entailment loss

In addition to the contrastive loss, we adapt an entailment loss [74, 139] to enforce partial order relationships between paired text and images. Ganea et al. [74] is more different

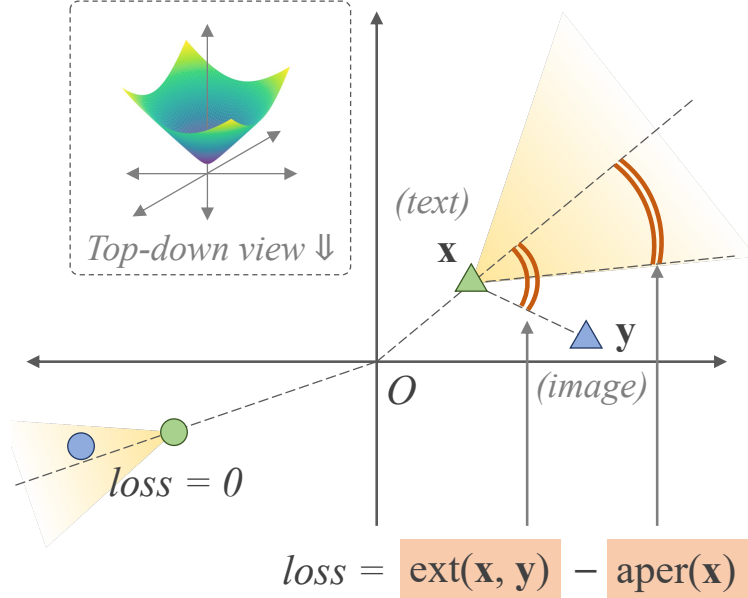


Figure 4.3: **Entailment loss (illustrated for \mathcal{L}^2)**: This loss pushes image embedding y inside an imaginary cone projected by the paired text embedding x , and is implemented as the difference of exterior angle $\angle Oxy$ and half aperture of the cone. Loss is zero if the image embedding is already inside the cone (*left quadrant*).

from ours since they parameterize their representations according to the Poincaré ball model. Le et al. [139] use this loss with a fixed $c = 1$, which we extend to handle arbitrary, learned curvatures.

Refer Figure 4.3 for an illustration in two dimensions. Let x and y denote the text and image embeddings of a single image-text pair. Note that the encoders only give x_{space} and y_{space} according to our parameterization. Corresponding x_{time} and y_{time} are calculated using Eqn. 4.3. We define an *entailment cone* for each x , which narrows as we go farther from the origin. This cone is defined by the half-aperture:

$$\text{aper}(\mathbf{x}) = \sin^{-1} \left(\frac{2K}{\sqrt{c} \|\mathbf{x}_{space}\|} \right) \quad (4.10)$$

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 y lies outside the entailment cone. For this, we measure the exterior angle $\text{ext}(\mathbf{x}, \mathbf{y}) = \pi - \angle Oxy$ as shown in Figure 4.3. This angle is computed as follows:

$$\text{ext}(\mathbf{x}, \mathbf{y}) = \cos^{-1} \left(\frac{y_{time} + x_{time} c \langle \mathbf{x}, \mathbf{y} \rangle_{\mathcal{L}}}{\|\mathbf{x}_{space}\| \sqrt{(c \langle \mathbf{x}, \mathbf{y} \rangle_{\mathcal{L}})^2 - 1}} \right) \quad (4.11)$$

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

$$\mathcal{L}_{entail}(\mathbf{x}, \mathbf{y}) = \max(0, \text{ext}(\mathbf{x}, \mathbf{y}) - \text{aper}(\mathbf{x})) \quad (4.12)$$

We provide exact derivations of the above equations for half-aperture and exterior angle in Appendix B.1. Overall, our total loss is $\mathcal{L}_{cont} + \lambda\mathcal{L}_{entail}$ averaged over each minibatch.

4.4 Experiments

Our main objective in the experiments is to establish the competitiveness of hyperbolic representations of MERU as compared to Euclidean representations obtained from CLIP-style models. To this end, we train models using large amounts of image-text pairs and transfer them to a variety of image classification and retrieval tasks.

4.4.1 Training details

Baselines. We primarily compare with CLIP [198], 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 [236], Conceptual Captions [30, 215], and LAION [210, 211]; notable examples are OpenCLIP [110], SLIP [176], DeCLIP [150], and FILIP [268]. We develop our CLIP baseline and train it using a *single* public dataset – RedCaps [51] – for easier reproducibility. Our smallest model trains using $8 \times$ V100 GPUs in *less than one day* and significantly outperforms recent CLIP re-implementations that use YFCC [176]. Refer Appendix B.3 for details about our CLIP baseline. Our implementation is based on PyTorch [187] and timm [255] libraries.

Models. We use the Vision Transformer [60] as image encoder, considering three models of varying capacity – ViT-S [35, 240], 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 [244] language model. We use the same byte-pair encoding tokenizer [213] 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 [176]. 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 [35, 96], 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 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 [162] 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 (≈ 20 epochs). The maximum learning rate is 5×10^{-4} , increased linearly for the first 4K iterations, followed by cosine decay to zero [161]. We use mixed precision [172] 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.

4.4.2 Image and text retrieval

CLIP-style contrastive 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 [34, 270], 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 and Fei-Fei [118]. We perform *zero-shot transfer*, without any additional training using these datasets. We *squeeze* images to 224×224 pixels before processing them through the image encoder.

Inference with MERU. We rank a pool of candidate image/text embeddings for retrieval in decreasing order of their Lorentzian inner product (Eqn. 4.1) with a text/image query

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

Table 4.1: **Zero-shot image and text retrieval.** Best performance in every column is highlighted in green. MERU performs better than CLIP for both datasets and across all model sizes.

embedding. Some transfer tasks like *open-vocabulary detection* [85, 276] may require calibrated scores, for them we recommend using the training procedure – compute the negative of distance (Eqn. 4.4), divide by temperature and apply a softmax classifier.

Results. In Table 4.1, we report $\text{recall}@_{\{5,10\}}$ of MERU and the reproduced CLIP baselines on COCO and Flickr benchmarks. Hyperbolic representations of MERU 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 model size (ViT-B/16 \rightarrow ViT-L/16) does not improve image retrieval for both, MERU and CLIP. We believe that image retrieval can be improved by using text embeddings (queries) of better quality – increasing the size of text encoder can alleviate this issue.

4.4.3 Image classification

Learning from language supervision allows CLIP to perform *zero-shot* image classification, wherein one may specify label sets as text queries [65] instead of using pre-defined ontologies [47, 173]. Classifier weights are obtained by embedding label-based queries (also called *prompts*) using the text encoder.

In this section, we evaluate MERU on 20 image classification benchmarks covering a wide variety of visual concepts. These are used by Radford et al. [198] and several follow-up works [150, 176, 268]. We use two open-source libraries to access these datasets –

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

Table 4.2: **Zero-shot image classification.** We train MERU and CLIP models with varying parameter counts and transfer them *zero-shot* to 20 image classification datasets. Best performance in every column is highlighted in *green*. Hyperbolic representations from MERU match or outperform CLIP on 13 out of the first 16 datasets. On the last four datasets (*gray* columns), both MERU and CLIP have *near-random* performance, as concepts in these datasets are not adequately covered in the training data.

tensorflow-datasets and torchvision². 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. [198]. We *ensemble* these multiple prompts by averaging their embeddings before lifting them onto the hyperboloid (Eqn. 4.9). See Appendix B.2 for details about datasets and prompts.

Results. Table 4.2 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 4.1), it does not come at the expense of precision [177]. 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

²tensorflow.org/datasets and pytorch.org/vision

		Embedding width				
		512	256	128	96	64
COCO <i>text</i> → <i>image</i>	CLIP	31.7	31.8	31.4	29.6	25.7
	MERU	32.6	32.7	32.7	31.0	26.5
COCO <i>image</i> → <i>text</i>	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

Table 4.3: **MERU for resource-constrained deployment.** We compare MERU and CLIP at different embedding widths on *zero-shot* classification and retrieval tasks (COCO recall@5 and ImageNet top-1 accuracy). MERU outperforms CLIP at lower embedding widths.

datasets have low coverage in RedCaps, like PCAM [246] containing medical scans, or SST2 [218] 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 [211] may yield meaningful trends.

4.4.4 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 [134]. 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 4.2, 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 [207] classification. Results in Table 4.3 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.

4.4.5 Ablations

In this section, 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 (Section 4.4.1), except having one difference each. Results are shown in Table 4.4 above.

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

Table 4.4: **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.

No entailment loss: We only use the contrastive loss for training this ablation. This effectively means setting $\lambda = 0$. 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. 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. Entailment loss is crucial for better structure and interpretability, as will be discussed in Section 4.5.

Fixed curvature parameter: Recall that we learn the hyperboloid curvature 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 [7, 120, 180].

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. 4.1) 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 numerical growth and hence improving numerical stability.

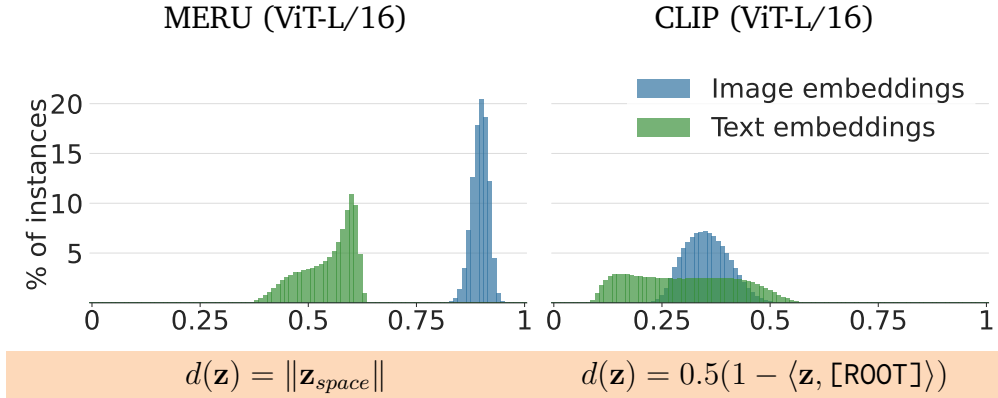


Figure 4.4: **Distribution of embedding distances from [ROOT]:** We embed all 12M training images and text using trained MERU and CLIP. Note that precise distance is not necessary for this analysis, so we compute simple monotonic transformations of distances, $d(\mathbf{z})$. MERU embeds text closer to [ROOT] than images.

4.5 Qualitative analysis

In this section, 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 MERU and CLIP models with the largest image encoders (ViT-L/16).

4.5.1 Preliminary: Root node embedding

Recall Figure 4.1 – 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 $2 \times 12\text{M}$ 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.

4.5.2 Embedding distances from the root node

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 (Figure 4.1). Figure 4.4 shows the distribution of embedding distances from [ROOT] – these distributions overlap for CLIP but are separated for MERU. The range of distributions in Figure 4.4 (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 the ring of images is relatively *thin*. This resembles the structure of the visual-semantic hierarchy – images only occupy *leaf nodes* whereas text occupies many intermediate nodes.

4.5.3 Image traversals

In a discrete tree, one can discover the *ancestors* of any node by performing shortest-path traversal to the *root node* [54]. We perform such traversals for images with MERU and CLIP. If the representation space has captured the visual-semantic hierarchy, then a shortest-path traversal from an image to [ROOT] should let us infer textual concepts that describe the image with varying levels of abstraction.

We traverse from an image to [ROOT] by interpolating $N = 50$ equally spaced steps along the geodesic connecting their embedding vectors. The embedding vector of every interpolated *step* is used as a *query* to retrieve the nearest neighbor from a set of text embeddings \mathcal{X} (also including [ROOT]).

Collecting image-text pairs for qualitative results: For this analysis, we require a set of images and text descriptions that describe images with varying levels of specificity. We collect images from pexels.com, a website that offers high-quality stock photographs with free and permissible usage terms. Images on this website are accompanied with rich textual metadata. We manually collect 60 random images along with their textual metadata; an example webpage is shown in Figure 4.5. We perform parts-of-speech tagging of all keywords using the RoBERTa [158] model (en-core-web-trf) from SpaCy [106] library, and only retain nouns and adjectives. These keywords are converted to captions by filling prompts – ‘a photo of {}.’ for nouns, and ‘this photo is {}.’ for adjectives. Overall, we get a total of 750 captions to create the set of their embeddings, \mathcal{X} .

Interpolating steps: MERU and CLIP have different methods for interpolation due to the difference in geometric properties of Euclidean and hyperbolic spaces. For CLIP, we perform linear interpolation between the L^2 normalized image embedding y and [ROOT]. This

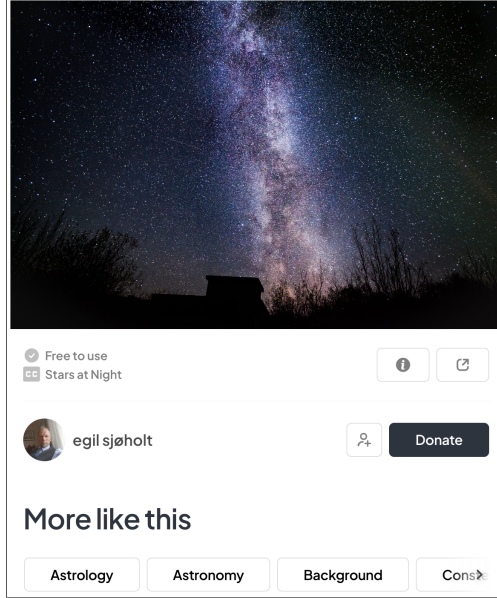


Figure 4.5: [pexels.com](https://www.pexels.com) webpage. We collect images and associated textual metadata (closed caption, CC and related keywords, ‘More like this’) from this website to create retrieval sets for the image traversal analysis.

operation can be performed using `torch.lerp` in PyTorch [187]. For MERU, we perform linear interpolation between encoder output y_{space} (before lifting it onto the hyperboloid) and 0. We then lift all the embeddings of interpolated steps onto the hyperboloid.

Nearest-neighbor text retrieval: This procedure is similar to our image retrieval evaluations. For CLIP, we select $x \in \mathcal{X}$ having the highest cosine similarity with the *step* embedding. For MERU, we find a subset $\mathcal{X}_e \subset \mathcal{X}$ of text embeddings that *entail* the given *step* embedding, *i.e.*, Eqn. 4.12 evaluates to 0 (note that [ROOT] entails everything). Then we select $x \in \mathcal{X}_e$ having the highest *Lorentzian inner product* with the *step* embedding.

Results: Figure 4.6 shows results with 8 selected images and captions from [pexels.com](https://www.pexels.com). At any given step, the caption associated with the retrieved text embedding x (or [ROOT]) is the retrieved nearest neighbor. We observed that multiple consecutive *steps* retrieve the same caption, so our results only display *unique* captions encountered during the traversal. CLIP seems to capture hierarchy to some extent, often retrieving very few (or zero) captions between image and [ROOT]. MERU captures it with much finer granularity, retrieving concepts that gradually become more *generic* as we move closer to [ROOT]. Figures 4.7 to 4.11 show results with the remaining 52 images. Appendix B.5 includes results using captions from the YFCC dataset [236] as a retrieval pool.






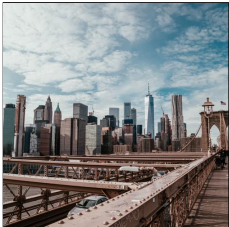
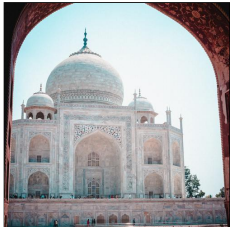
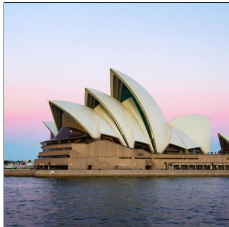
							
MERU	CLIP	MERU	CLIP	MERU	CLIP	MERU	CLIP
<i>a bengal cat sitting beside wheatgrass on a white surface</i>	<i>a bengal cat sitting beside wheatgrass on a white surface</i>	<i>white horse</i>	<i>white horse</i>	<i>photography of rainbow during cloudy sky</i>	<i>phenomenon</i>	<i>retro photo camera on table</i>	↓
<i>bengal</i>	↓	<i>equine</i>	↓	<i>rainbow</i>	↓	<i>fujinomiya</i>	↓
<i>cat</i>	↓	<i>equestrian</i>	↓	<i>phenomenon</i>	↓	<i>vintage</i>	↓
<i>domestic</i>	↓	<i>beauty</i>	↓	<i>rural</i>	↓	<i>style</i>	↓
[ROOT]	[ROOT]	[ROOT]	[ROOT]	[ROOT]	[ROOT]	[ROOT]	[ROOT]
							
MERU	CLIP	MERU	CLIP	MERU	CLIP	MERU	CLIP
<i>avocado toast</i>	<i>avocado toast</i>	<i>brooklyn bridge</i>	<i>photo of brooklyn bridge, new york</i>	<i>taj mahal</i>	<i>taj mahal through an arch</i>	<i>sydney opera house</i>	<i>sydney opera house</i>
<i>healthy breakfast</i>	<i>delicious</i>	<i>new york city</i>	<i>new york city</i>	<i>monument</i>	<i>travel</i>	<i>opera house</i>	<i>opera house</i>
<i>delicious</i>	↓	<i>city</i>	<i>new york</i>	<i>architecture</i>	<i>inspiration</i>	<i>holiday</i>	<i>gift</i>
<i>homemade</i>	↓	<i>outdoors</i>	↓	<i>travel</i>	↓	<i>day</i>	<i>beauty</i>
<i>fresh</i>	↓	<i>day</i>	↓	<i>day</i>	↓	[ROOT]	[ROOT]
[ROOT]	[ROOT]	[ROOT]	[ROOT]	[ROOT]	[ROOT]	[ROOT]	[ROOT]

Figure 4.6: **Image traversals with MERU and CLIP.** CLIP retrieves overall 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*. (2) MERU has higher recall of concepts than CLIP, e.g., *homemade*, *city*, *monument*. (3) MERU shows systematic text→image entailment, e.g., *day* entails many images captured in daylight.

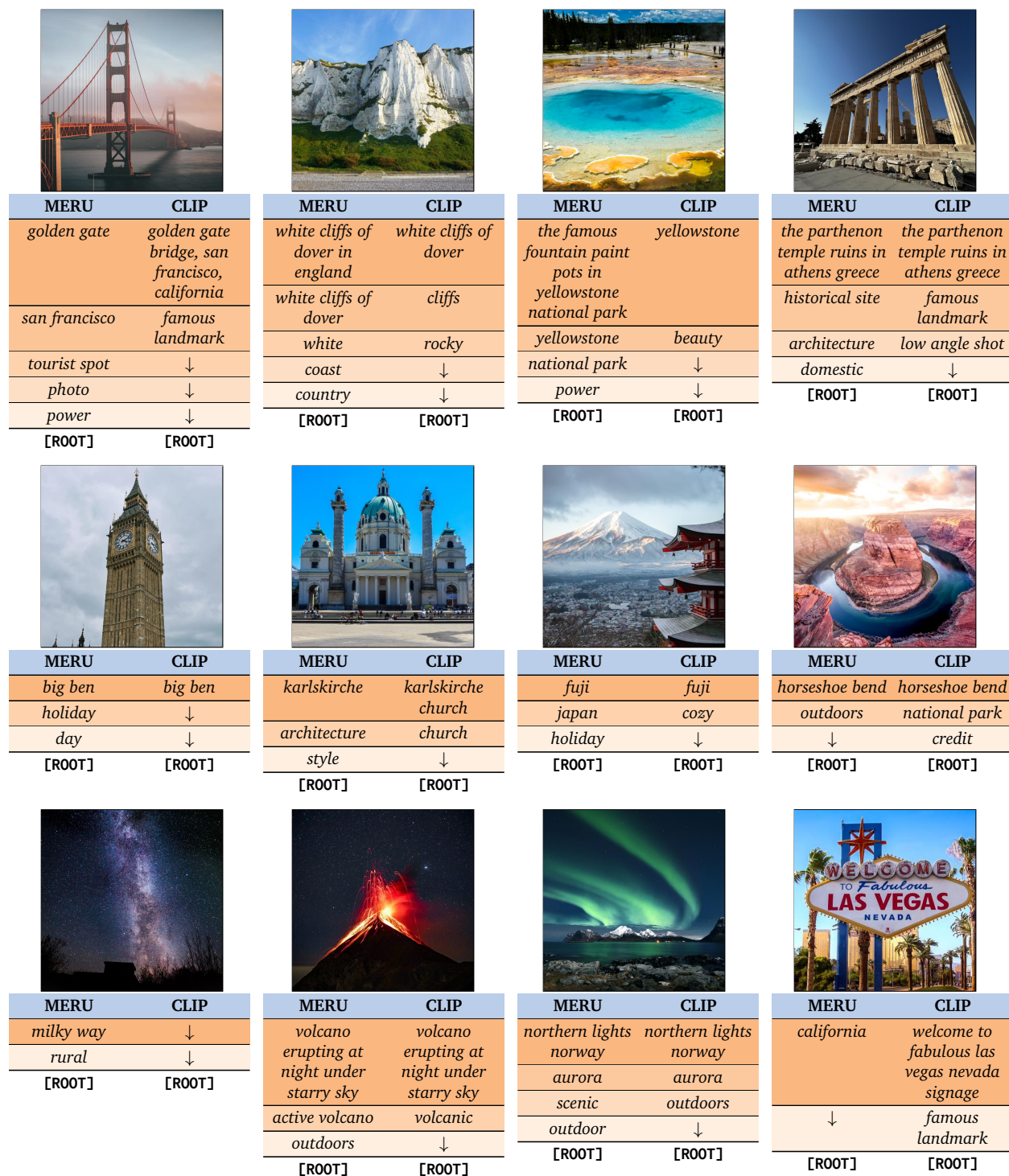


Figure 4.7: Image traversals with MERU and CLIP (locations and landmarks). Retrieved captions are sourced from pexels.com metadata. MERU captures a more systematic and fine-grained visual-semantic hierarchy than CLIP.

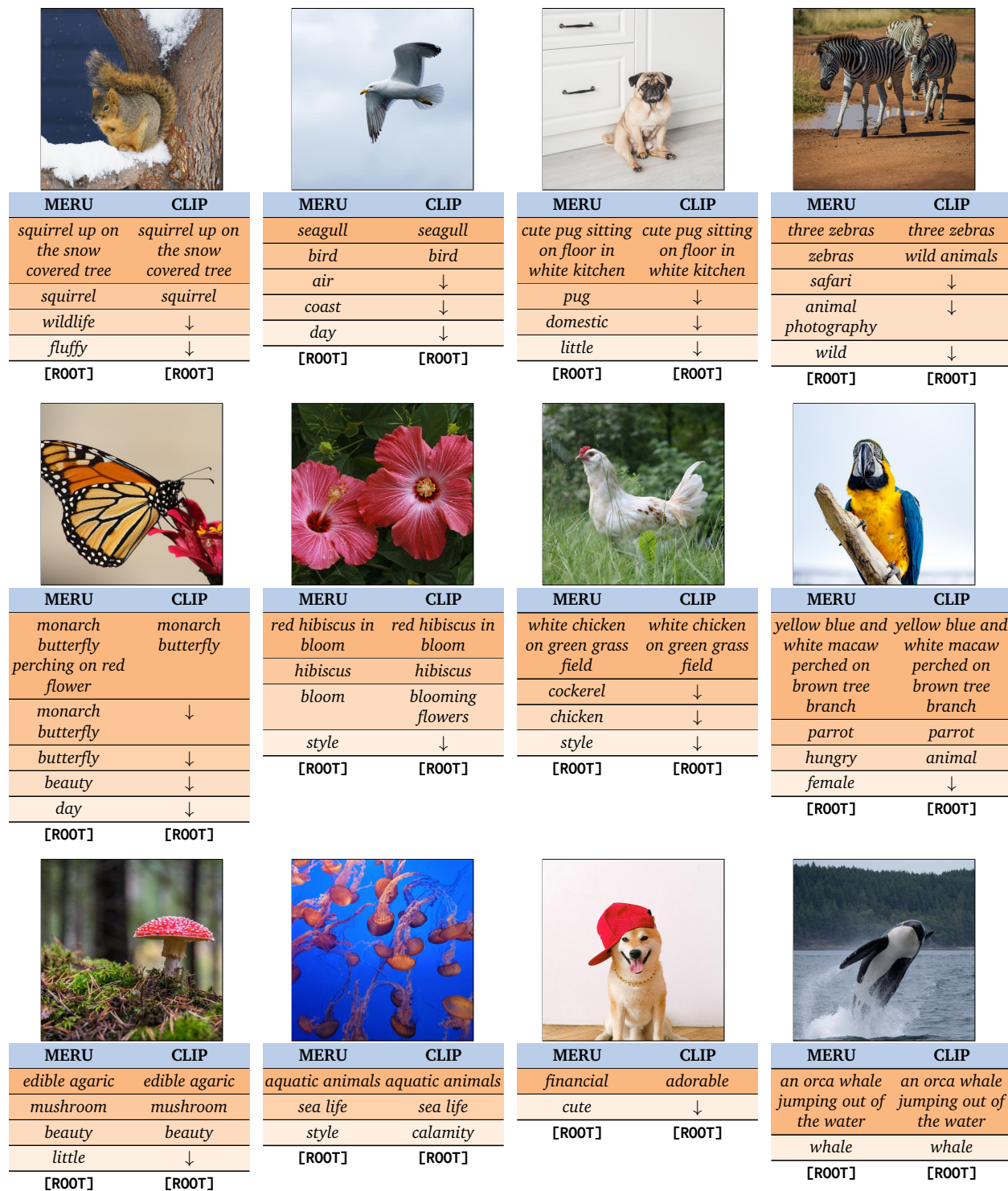


Figure 4.8: Image traversals with MERU and CLIP (flora and fauna). Retrieved captions are sourced from [pexels.com](https://www.pexels.com) metadata. MERU captures a more systematic and fine-grained visual-semantic hierarchy than CLIP.



Figure 4.10: **Image traversals with MERU and CLIP (objects and scenes).** Retrieved captions are sourced from [pexels.com](https://www.pexels.com) metadata. MERU captures a more systematic and fine-grained visual-semantic hierarchy than CLIP.




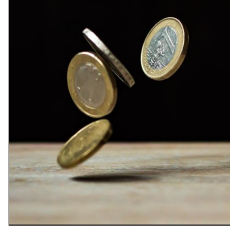
			
MERU	CLIP	MERU	CLIP
turned on floor lamp near sofa on a library room	bookshelves	pineapple	ripe pineapple on gray rock beside body of water
books	↓	inspiration	pineapple
bookshelves	↓	health	calamity
comfort room	↓	little	↓
cozy	↓	[ROOT]	[ROOT]
style	↓		
[ROOT]	[ROOT]		
		MERU	CLIP
		cockatiel	cockatiel
		female	↓
		[ROOT]	[ROOT]
		MERU	CLIP
		currency	euro
		simple	revenue
		[ROOT]	[ROOT]

Figure 4.11: **Image traversals (objects and scenes)**. Retrieved captions are sourced from pexels.com metadata. MERU captures a more systematic and fine-grained visual-semantic hierarchy than CLIP.

4.6 Related work

Visual-language representation learning. Soon after the initial success of deep learning on ImageNet [131], deep metric learning [220, 221] was used to learn vision-language representations in a shared semantic space [70, 118]. The motivations at the time included the possibility of improving vision models [70], enabling zero-shot learning by expressing novel categories as sentences [65, 70], and better image-text retrieval [118, 270]. Another line of work proposed learning visual models from language supervision via objectives like textual n-gram prediction [142], or *generative* objectives like masked language modeling [23] or image captioning [50].

More recent approaches like CLIP [198] and ALIGN [114] use contrastive metric learning to pre-train Vision Transformers [60] 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 (Figure 4.1) underlying images and text. Our results (Section 4.4) 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 [247] 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 incorporating distance-based contrastive learning is crucial to obtain better performance on both, image classification and retrieval. Thus, we opt for adapting the currently successful CLIP-style contrastive learning recipe and add our entailment objective in conjunction. This design choice helps us obtain the desired structure in the representation space.

For NLP and knowledge graph embedding applications, several approaches embed partially ordered data [11, 45, 74, 178, 248] or discover ordering from pairwise similarities [139, 179, 239]. This chapter builds upon both of these lines of work, since we impose structure *across* modalities, but order also emerges *within* modality (Figure 4.6).

Hyperbolic representations in computer vision. Khrulkov et al. [120] learn hyperbolic image embeddings using image-label pairs, while Atigh et al. [7] study image segmentation by utilizing hyperbolic geometry. More recently, Ermolov et al. [66] and Ge et al. [75] extend standard contrastive self-supervised learning framework [95, 257] 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 [198].

4.7 Conclusion

In this chapter, we focused on a practical aspect of learning language-supervised representations – understanding and interpreting the distribution of concepts underlying millions of images and text. We proposed learning hyperbolic representation with MERU to capture the visual-semantic hierarchy underlying image-text datasets. MERU is competitive or more performant than approaches that learn Euclidean representations (like CLIP). It does so along with capturing hierarchical knowledge which allows one to make powerful inferences such as reasoning about images at different levels of abstraction. Beyond this, our model also provides clear performance gains for small embedding dimensions (which are useful in resource-constrained settings). We hope that our contributions catalyze progress in learning useful representations from large amounts of unstructured data.

Future work. In this *scaling* era, we are seeing rapid progress with large multi-modal models trained using millions (or even billions) of image-text pairs. The quality and concept distribution of training data play a vital role in the efficacy of these models. Such training data is becoming increasingly opaque and black-box due to its unprecedented

scale. We believe that the time is ripe to revisit the unreasonable effectiveness of data in deep learning [89, 227]. Modeling hierarchies can help uncover higher-order relationships beyond basic data statistics. As a concrete example, Figure 4.1 “so cute <3” is an extremely generic caption and does not the *precise* details in images. Such captions add noisy supervision in contrastive loss by making false negative pairs with many images in the batch. Image traversals with MERU Figure 4.6 can discover such noisy captions. ML practitioners can filter or re-caption such training images to improve dataset quality and train subsequent models for improved performance.

Limitations. Our work is not without limitations. MERU yields hyperbolic representations that excel at zero-shot retrieval and image classification tasks, the linear probe evaluations in the Table B.4 show that the underlying Euclidean representations from the image encoder of MERU underperform CLIP. Future work could explore MERU’s transferability to other tasks that involve few-shot learning or full-model fine-tuning, which is also beyond the scope of this chapter. Finally, while we provide ample qualitative analysis of image traversals, future work should explore more systematic ways to evaluate the hierarchical knowledge captured by vision-language models.

objects of interest, *e.g.*, through bounding boxes or segmentation masks, and classify each object using a specified label set. There has been a lot of progress in trying to solve these problems independently. Recently, works in interactive segmentation [123] have pushed the limits to localize objects in novel images with unseen classes but do not predict object semantics. On the other hand, we have also seen works like CLIP [198] can learn large vision-language models from image-text paired data. These models are good at understanding semantics in case of unseen classes directly from text supervision. Jointly solving semantics and locations still requires using two-stage architectures with ConvNet backbones [92] or transformer backbones [60] but is limited to datasets like COCO [153] and LVIS [86] and training such detectors is very data-inefficient.

In this chapter, we will revisit the object detector designs and aim to incorporate large vision models like SAM and CLIP for efficient transfer learning. With the advent of models on web-scale data, it is natural to train bigger and stronger models as it is easy to collect data such as image-text pairs, video-text pairs etc compared to collecting data such as extensive masks, bounding boxes and class labels. On the other hand, it is much easier to collect loose mask annotations without explicit ontology allowing the scaling of interactive segmentation models. As these models continue to scale, we must reconcile the efforts in two directions to approach the task of object detection without re-inventing the wheel.

We propose a simple object detector named SCAM, short for Segment and Classify Anything Model. SCAM is composed of two promptable models for segmentation and classification respectively – SAM and CLIP. Our key insight is that models could be used to prompt each other for instance segmentation with minimal human guidance: only the classes of interest. We use CLIP’s embeddings to both prompt SAM and classify SAM’s predicted masks. Our setup keeps large parts of CLIP and SAM *frozen* and retains the full functionality of these models at initialization. This deliberate design choice allows using SCAM out-of-the-box for segmentation in challenging low-data regimes like training-free segmentation, and training with unlabeled masks.

In our experiments, we will train SCAM using popular object detection benchmarks – COCO and LVIS, along with reporting its zero-shot transfer performance on these benchmarks without any additional training. Under fair comparisons holding the image backbone and training data constant, we find that SCAM outperforms prior object detector designs on all considered image segmentation benchmarks.

5.2 Related Work

We build on a long line of work that develops simple and scalable models for object detection and segmentation.

Object detectors. Object detection is a fundamental vision task that involves localizing and classifying a candidate set of objects in an image. Modern object detectors follow a general design based on the pioneering R-CNN [81] and its faster variants [80, 204]. These models follow a *two-stage*, *anchor-based* and *region-based* design. Follow-up works introduce different type of variants like one-stage [155, 238, 289], anchor-free [238, 253, 254], and more recently query-based universal segmentation models [25, 37, 38].

While different in their design, all these models can be characterized as a combination of: (i) generic image backbone like ConvNet [91, 131, 159, 261] or ViT [60], and (ii) detection-tailored modules like FPN [154], and RoI heads [92]. With an exception of few studies that train the entire detector from scratch [78, 93], nearly every work uses an image backbone pre-trained using ImageNet [47] and fine-tunes other components. In all these designs, the task-specific components are attached in a way that *breaks* the alignment of backbone features with their label space, or have backbones that are not aligned with a semantic label space (*e.g.*, self-supervised MAE [96]). Moreover, before the advent of SAM [123], all these models learn segmentation-specific modules *from scratch*. With SCAM, we de-facto use CLIP and SAM and deliberating opt for a simple design for efficient downstream transfer.

Open-vocabulary object detection and segmentation. This task [276] is a successor to the challenging *zero-shot detection* [12], and requires models to detect (and segment) object classes for which masks are not available in the training data. With the advent of vision-language models like CLIP [198] and ALIGN [114], open-vocabulary object detection has broadened to rigorous empirical study. From a scaling perspective, this task is appealing as the data requirements are *lighter* than traditional, closed-set detection.

CLIP’s *promptable* classification allows building detectors with text classifier weights from pre-trained text encoder of CLIP, facilitating seamless transfer. Several works utilize CLIP-style models in their design for knowledge distillation [85] or to generate pseudo region-label pairs for detection pre-training [286]. Recent works further simplify this by using CLIP image encoders to classify *image crops* [152, 274] or as detector backbones [90, 133, 144, 264, 273].

Unlike SCAM, these approaches still require a large amount of labeled detection data

as they do not preserve the alignment between the pre-trained image-text representations. Many of these works only tackle bounding box-based detection. We design SCAM for data-efficiency, and our experiments study downstream transfer in both, *zero-shot* as predicting pixel-precise masks for novel classes is challenging [20]. We demonstrate how SCAM can be adapted to low-data regime by training lightweight modules requiring fewer masks and lower compute budgets than prior work.

5.3 Approach

This work proposes a simple framework for object detection and instance segmentation by merging two disparate research efforts on building promptable vision foundation models for zero-shot image segmentation (SAM [123]) and classification (CLIP [198]). Object detection involves two sub-tasks: (1) localize all object instances of interest in an image and (2) classify each instance into a set of object classes of interest. We decouple these sub-tasks and *offload* them to the underlying components (SAM and CLIP) respectively to build our SCAM.

Desiderata. Our design philosophy is to preserve the existing capabilities of pre-trained models to enable fast downstream transfer. This would allow SCAM to either be used out-of-the-box without additional training or to be rapidly steered toward a target task with minimal data and compute requirements. This departs from existing detectors [25, 92] that initialize, and train, new modules on top of image backbones and *break* the alignment with learned label space from pre-training, or opt for pre-trained backbones whose representations are not aligned with a textual representation space [151]. While we operationalize this with the currently available SAM and CLIP models, our design applies to any promptable segmenter and classifier.

5.3.1 Model architecture

As shown in Figure 5.2, SCAM’s design is simple and modular, can be broken down into four sub-modules: (i) the *backbone* extracts dense embeddings from input images, (ii) the *prompter* proposes a set of points for the *segmenter*, (iii) the *segmenter* predicts class-agnostic masks based on point prompts, and (iv) the *classifier* assigns a class label to masks from the segmenter. Now we introduce each of the components in detail while making minimal assumptions about the underlying structure of these modules, which makes our design accommodating to a variety of architectures.

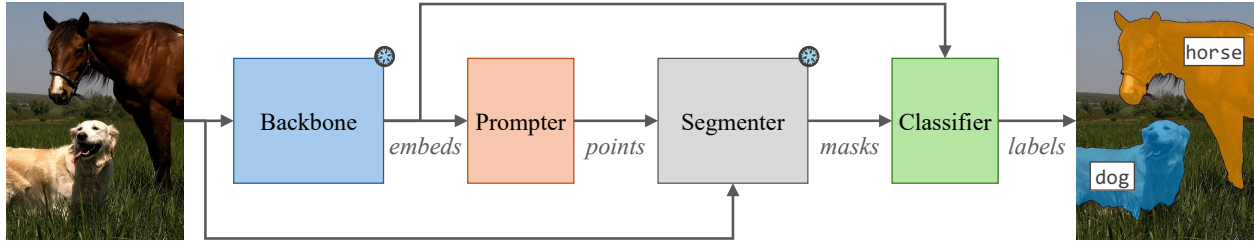


Figure 5.2: **SCAM Design:** Image backbone extracts embeddings from an input image. The prompter uses backbone embeddings to propose a set of pixel locations for our visual concepts of interest. The segmenter uses the image and pixel prompts to predict a set of binary masks. Finally, the classifier performs mask classification using backbone embeddings and masks from the segmenter to predict class labels. Our backbone and segmenter are frozen throughout while we optionally train the light weight prompter and classifier for various data regimes.

Backbone. The backbone is a deep network that encodes an image as a dense grid of image embeddings. According to prevailing practices [78, 151], we provide 1024×1024 image inputs and obtain spatial embeddings of size 64×64 (stride = 16). This amounts to using a backbone that downsamples the image with an overall scale of $1/16$, e.g., ViTs [60] with patch size of 16, or the initial few *stages* of convolutional (hierarchical) models until they obtain embeddings of $1/16$ scale ¹.

In our design, we partition the convolutional CLIP image encoder and include the initial layers in the backbone. As per the requirement stated above, we remove all layers including and after the downsampling layer in the last convolutional stage. We briefly experiment with ViT-based CLIP models, which we found to underperform their ConvNet-based counterparts – see Appendix C for discussion.

Prompter. The prompter is a lightweight, fully convolutional module that inputs the image embeddings from the backbone and predicts a heatmap with values in $[0, 1]$. High values indicate the presence of an object of interest. Pixel locations with high *objectness* are prompted to the segmenter. Intuitively, the prompter ensures high recall over object regions of interest while improving over the worst-case runtime of selecting all possible points as prompts to the segmenter. Functionally, this prompter is equivalent to the Region Proposal Networks, however, it uses points as the fundamental substrate for detection instead of anchor boxes.

¹A typical convolutional model downsamples the image by $1/32$

Segmenter. The segmenter is a promptable segmentation model, responsible for predicting label-agnostic masks based on the prompter outputs. We use SAM [123], specifically the largest model with ViT-H image encoder for the highest quality masks. While SAM can predict masks conditioned on a combination of points and boxes, we only use the points obtained from the *prompter* as single-point prompts and obtain three masks (*multi-mask* mode). We experimented with bounding box prompts and found the point prompts to lead to better results. Our modular design is compatible with any other promptable segmentation models [33, 157, 219] beyond SAM.

Classifier. The classifier inputs the dense image embeddings from the backbone, crops region embeddings using masks from the segmenter, and classifies each mask (region embedding) as one of the labels provided as prompts by the user. This module contains two sub-components: (1) the last few layers of the CLIP image encoder which were excluded from the backbone, and (2) a single weight matrix comprising embedding vectors obtained by encoding label prompts using the CLIP text encoder.

The backbone and classifier collectively consume the entire pre-trained CLIP model, and introduce *no additional trainable parameters*. Since we preserve the *forward pass* of CLIP, we obtain *zero-shot* classification out of the box.

5.3.2 Modeling Decisions

Now, we discuss how current prevalent model architectures such as [92] do not lend themselves well in data-starved settings due to their task-specific modules, and how our modification help solve these limitations

Data starved settings. Transfer learning for object detection has always been heavily data-demanding – obtaining domain-specific labeled masks is not always cheap or feasible. We designed SCAM with a specific focus to reduce the need for large amounts of labeled masks to bootstrap a reasonable performing detector for any specific detection domain. Existing object detectors, *e.g.*, those based on the popular Mask R-CNN architecture, are composed of modules that are task-agnostic and task-specific. Commonly, the image backbone is the main task-agnostic component, which represents the bulk of the detector and is pre-trained using other types of data and supervision that are usually more scalable than detection data. Until now, the task-specific components were initialized from scratch and designed in a manner that made it impossible to use the detector without being trained with a detection dataset.

Task-specific modules. Our choice using SAM as a segmenter in our model departs from the current object detector design [37, 92] that couples segmentation and classification tasks and tackles them with task-specific modules attached on backbones. With SCAM, we eliminate these task-specific components and replace them with a promptable segmentation model like SAM, that can scale with arbitrarily large amounts of unlabeled, class/task-agnostic masks.

Different from existing detectors, our backbone omits any mechanism to obtain multi-scale image embeddings *e.g.*, a feature pyramid network (FPN). This deliberate choice allows us to further eliminate many task-specific modules and hyperparameters (*e.g.*, RPN, localization-specific ROI heads). In fact, in our preliminary experiments, these components bring very few empirical benefits for SCAM. We hypothesize that multi-scale feature pyramids are beneficial when learning pixel-wise segmentation *from scratch*, but they are less important with SCAM as segmentation is completely handled by SAM. Moreover, the *ambiguity-aware* design of SAM naturally provides masks of different scales.

In summary, we opt to omit most of the task-specific components from prior detector designs in favor of simplicity – using stronger backbones and segmenters can easily recover the benefits brought by task-specific modules.

5.3.3 Training and Inference

SCAM is readily usable without additional training, owing to our careful design. However, its capabilities are sub-par due to the train-test domain mismatch for CLIP. Here, we prescribe a training recipe for the prompter and optionally, to fine-tune the classifier.

Training the prompter. One may use a few *unlabeled masks* depicting objects of interest to steer the prompter towards selecting a subset of points from the dense 128×128 point grid – higher precision in selecting *foreground* points would improve runtime (less *forward passes* through segmenter).

We train the prompter to predict a score $\in [0, 1]$ for each feature in the dense image level feature grid. We require our model to only generate candidates for pixels that belong inside a mask. SAM tends to predict better masks for point prompts that lie closer to object centers, so we use a mask-based *centerness* [238] as *ground-truth* regression targets.

Specifically, we implement this as the nearest distance to the mask boundary – by computing Euclidean distance transform (EDT) [206] over image masks. We use a pixel-wise *mean-squared error* loss, $L_{\text{prompt}} = ||f_{\phi}(I) - EDT(M)||^2$. $M[p]$ is the mask value at pixel p , for any pixel location p in the image of size $H \times W$ the EDT function computes the

distance to the nearest background pixel.

At inference, we find *peaks* in the predicted heatmap of logits using a 3×3 max-pooling kernel to sample points closer to object centers. In Figure 5.3 (a), we show the predicted logits by our prompter for a single input image. Regions with high intensity indicate a higher likelihood of good candidate point prompts. These sampled points serve as the input for the interactive segmentation module. Further details on training this module are in Appendix C.1.3.

Training Classifier. Note that training the classifier is optional, however, using a few labeled masks of downstream tasks allows the classifier to model the background class and discard proposals in the background regions. We train the classifier similar to training an ROI head in Mask R-CNN [92]. Specifically, we consider masks generated from the *segmenter* with an IoU ≥ 0.5 with the ground-truth mask. We assign this mask the corresponding label from the dataset otherwise treat it as background. During training, we randomly sample up to 64 proposal masks per image and train the classifier with standard cross-entropy loss. We weigh the *background* class by 25% to account for class imbalance.

Mask-based post-processing. The strategy to use different models to generate masks and labels generates a different type of redundancy in mask predictions that requires post-processing as we do not have access to bounding boxes like in RPN style designs [80, 92, 204]. During post-processing, firstly, we apply non-maximum suppression (NMS) using masks instead of bounding boxes. Using Mask NMS is equivalent in most cases, but it helps in case of occlusions (*e.g.*, giraffes standing behind each other), or when SAM predicts masks with spurious islands (yielding vastly different boxes). Secondly, SAM often segments object-subparts such as the eyes and nose of a bear (Figure 5.3) which are often so small that NMS cannot remove them. For such scenarios, we suppress a model’s mask prediction if at least 80% of its area is occluded by a larger mask of the same class, that is predicted with higher confidence.

5.4 Experiments

We evaluate SCAM on object detection and instance segmentation. Our experiments evaluate the transfer efficiency and downstream performance of our model. We are primarily interested in understanding how well our model performs when provided with different types of data describing the target class. This can differ from just knowing the names of

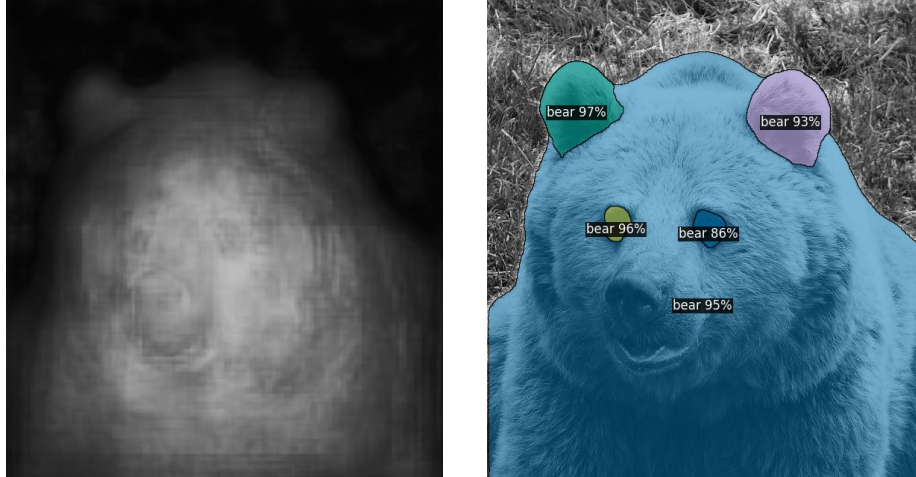


Figure 5.3: **Mask-based post-processing for SCAM.** **Left:** We show predicted logits for the image from our prompter. Bright colors indicate pixels deep inside a mask, while darker regions are near or outside the object boundary. This allows us to sample a few high-quality point prompts for the segmenter. **Right:** SAM generates subpart masks that get classified as the full object. Since those masks wholly overlap with the full-object mask, they can be easily detected. We propose a sub-mask suppression (SMS) technique to efficiently find and suppress these masks.

the classes to detect to being provided with labeled masks of those classes. Specifically, our experiments study the following questions:

1. How well does SCAM perform when it is just prompted with class names?
2. How well does SCAM perform when provided with unlabeled masks?
3. How well does SCAM perform when provided with a small number of labeled masks?

5.4.1 Zero-Shot Transfer

We first study the performance of SCAM in the zero-shot setting where we only know the names of the classes of interest. Since our design preserves the *forward pass* logic of the underlying components, we can directly apply it to a variety of segmentation tasks. Note that existing object detectors like Mask R-CNN [92], DETR [25], and Mask2Former [37] do not allow seamless zero-shot image segmentation as they rely on several segmentation-tailored, randomly initialized components on top of pre-trained backbones.

Model design: We use a modified version of SCAM for zero-shot experiments by removing all trainable components. We replace the FPN from the classifier and instead only use single-scale image embeddings from the backbone. Instead of a learnable prompter, we use SAM’s *segment everything* setup. This prompts the models with a fixed, image-independent

Backbone	Model	Runtime (sec/img)	COCO [153]		OID [16]	LVIS [86]			
			AP	AP ₅₀	AP ₅₀	AP	AP _r	AP _c	AP _f
ConvNeXt Base	Baseline	4.79	15.3	22.4	22.9	12.8	17.6	14.0	9.4
	SCAM	4.35	21.9	32.7	36.4	15.9	19.4	16.9	13.3
ConvNeXt Large	Baseline	5.00	16.8	24.6	26.9	16.4	24.0	17.9	11.3
	SCAM	4.58	24.8	37.9	41.8	18.9	22.5	20.5	15.5
ConvNeXt XXLarge	Baseline	5.66	18.3	26.6	27.0	18.7	26.2	20.4	13.4
	SCAM	5.52	25.8	39.1	41.1	21.2	25.6	23.0	17.2

Table 5.1: **Zero-shot transfer with SCAM.** We test experiment with our SCAM model three different datasets COCO [153], OpenImages [16] and LVIS [86]. The baseline method suffers from consistent errors due to background, and sub-parts. SCAM achieves substantial performance gains due to Mask NMS and subpart suppression across three different backbones all AP metrics. We show that with bigger backbones the performance improves. Running inference on SCAM and baselines in zero-shot mode requires ≈ 5 s per image.

point grid containing 64×64 points arranged uniformly on a (padded) input image of 1024×1024 pixels. We obtain three masks per point from SAM, resulting in up to 12.3K masks per image. Since the masks are not unique, we apply lightweight filtering following Kirillov et al. [123] by only retaining masks with a predicted IoU score greater than 0.7 and stability score greater than 0.9. Finally, we remove near-duplicates by performing non-maximum suppression with IoU threshold = 0.95. The filtering removes 78% of the masks, on average.

Baseline: Inspired by the R-CNN family of detectors [80, 81, 204], we design a baseline where SAM is used to propose potential regions and CLIP is used to classify them. Liang et al. [152] used a similar strategy where a MaskFormer trained on unlabeled COCO masks is used to generate masks and CLIP is used to classify the masked images. This strategy is very slow as it requires a forward pass of the model for each considered mask. We take inspiration from Fast R-CNN [80] and amortize the classification cost by applying region-of-interest (RoI) pooling on CLIP’s intermediate features. In this case, only the final layer of CLIP is applied for each segment, resulting in a drastic improvement in speed with minimal impact on performance.

Results and discussion. We observe that the baseline suffers from consistent errors involving backgrounds and object sub-parts as shown in Figure 5.4. Standard segmentation methods rely on a region proposal network that matches the granularity of the target class; *e.g.*, objects in COCO or Pascal, backgrounds in COCO-Stuff, or parts in COCO-Parts. Hence, while the region proposals may be agnostic to object classes, they are tuned

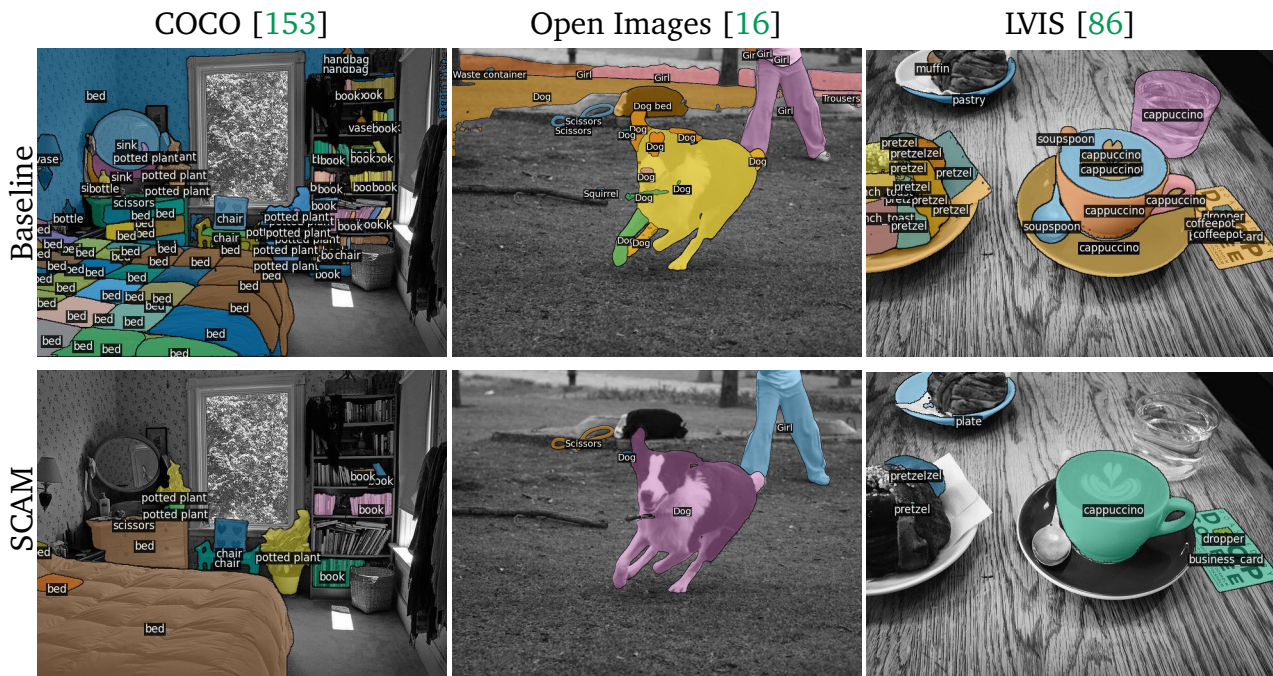


Figure 5.4: **Zero-shot results (qualitative)**. We observe that the baseline approach suffers from over-segmentation. Our test-time improvements with SCAM are effective in reducing these over-segmentations and producing reasonable outputs. *Note: Models receive RGB images, grayed here for better viewing.*

to a specific granularity. In contrast, SAM generates masks at different granularity levels by design. Such masks will also have a high IoU confidence since they are *valid* masks that match SAM’s training data; this makes it difficult to filter them out without human guidance. We also find that CLIP does not allow us to filter such masks easily as the features pooled for different mask granularity are often similar; *e.g.*, a bear’s face vs. its whole body as shown in Figure 5.3.

SCAM achieves much better segmentation performance even without fine-tuning. We attribute this gain to two specific design choices: (1) moving from boxes to masks and (2) sub-mask suppression. While feature pooling and non-maximal suppression are often computed using bounding boxes, we assert that pixel-precise masks are more accurate representatives of objects (especially thin objects), as compared to bounding boxes. Hence, we use masks for both feature pooling and non-maximal suppression (NMS).

While mask-based operations are more computationally expensive, efficient implementations make this increase negligible. Furthermore, mask NMS allows us to suppress more masks which helps offset the increased computational cost. As a result, moving to masks results in a net speedup of 0.66 sec/image as well as an improvement of 4.6 AP. Our second improvement comes from suppressing the sub-masks shown in Figure 5.3. This

Model Configuration (ConvNeXt-XXLarge)	Runtime (sec/img)	COCO	
		AP	AP ₅₀
Baseline	5.66	18.3	26.6
+ Masked Average Pooling	5.69 ^{+0.03}	22.6 ^{+4.3}	32.8 ^{+6.2}
+ Mask NMS	5.00 ^{-0.66}	22.9 ^{+4.6}	33.9 ^{+7.3}
+ Sub-Mask Suppression	5.01 ^{-0.65}	24.7 ^{+6.4}	36.8 ^{+10.2}
+ 24×24 RoI Align (SCAM)	5.52 ^{-0.14}	25.8 ^{+7.5}	39.1 ^{+12.5}
Oracle (uses GT boxes)	0.93	47.3	70.3

Table 5.2: **Zero-shot transfer ablations.** We ablate our design choices for SCAM on the COCO dataset [153]. Using our well-designed changes our method shows a performance improvement of 7.5 points over a naive standard baseline. We also show the upper bound segmentation performance of our design where we assume access to a perfect detector and prompt our model with GT bounding boxes.

minimally affects run-time while resulting in a further improvement of 1.8 AP. Finally, to further leverage our use of masks for pooling, we increase the RoI align size to sample more feature points around the mask. Taken together, our inference techniques result in a 7.5 improvement in AP and a speedup of 0.14 sec/image. While our zero-shot inference is too slow to be of practical use, it showcases the strong performance that could be achieved by intelligently combining existing models.

5.4.2 Transfer with Unlabeled Masks

While SCAM demonstrates strong zero-shot performance, it suffers from slow inference runtime, as the classifier encounters hundreds (or thousands) of masks from SAM.

We observe that most of these point prompts result in irrelevant masks, resulting in most of the compute being wasted. Ideally, we would only prompt SAM with relevant point prompts instead of the full point grid allowing us to significantly reduce the number of considered masks. In this section, we train the *prompter* module using unlabeled object masks, as described in Section 5.3.3.

and in Section 5.4.2 comparing performance to zero-shot exps from Section 5.4.1. We observe that when we train our lightweight prompter on all COCO masks, our performance improves slightly as compared to the zero-shot results as our points are steered toward COCO objects and ignore distractors, resulting in better accuracy and much faster runtime. Across all image backbones, training the prompter with unlabeled masks improves our AP while providing significant boosts in inference speed. Training the largest of the models, ConvNeXt-XXL [159], takes 17 hours on $2 \times$ NVIDIA A40 GPUs.

Backbone	Model	Runtime (sec/img)	COCO [153]	
			AP	AP ₅₀
ConvNeXt Base	SCAM (zero-shot)	4.35	21.9	32.7
	SCAM	0.79	22.2	32.9
ConvNeXt Large	SCAM (zero-shot)	4.58	24.8	37.9
	SCAM	1.07	25.7	38.8
ConvNeXt XXLarge	SCAM (zero-shot)	5.52	25.8	39.1
	SCAM	1.18	26.0	39.6

Table 5.3: **Training with unlabeled masks.** Transferring SCAM with unlabeled masks is beneficial for inference runtime ($5\times$ speedup). Training the prompter with unlabeled masks from the COCO dataset results in point prompts that are specific to the domain, and hence improves performance.

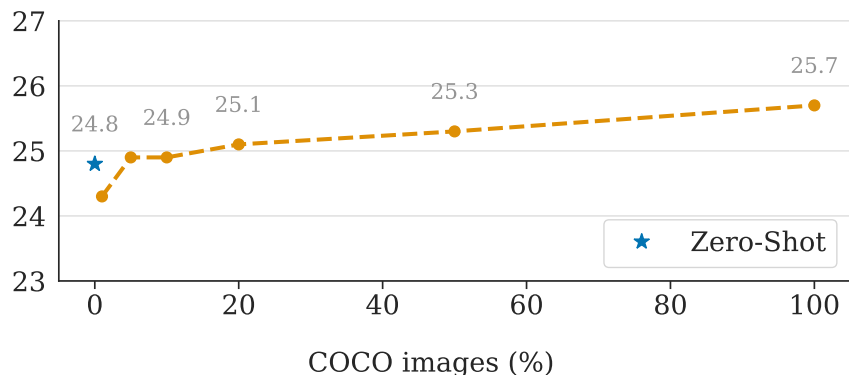


Figure 5.5: **Transferring SCAM using unlabeled masks (low data regime).** We observe the impact of training our prompter using subsets of COCO. We show that the prompter model is robust to the amount of data due to its lightweight design. The performance drop from 100% to 5% data is ≤ 1 AP.

Transfer in low data regime. In Figure 5.5, we report SCAM performance on COCO after training only the *prompter*, using different random subsets of COCO. We now experiment with training the prompter using unlabeled masks covering random subsets of images from COCO [153]. Similar to the previous setup, we freeze the backbone, segmenter, and classifier. We conduct experiments by randomly sampling 1%, 5%, 10%, and 50% of COCO masks with 5 different random seeds per sample. Our results, shown in Figure 5.5, show that even with as little as 5% data, our prompter is competitive with a model trained with the entire COCO dataset.

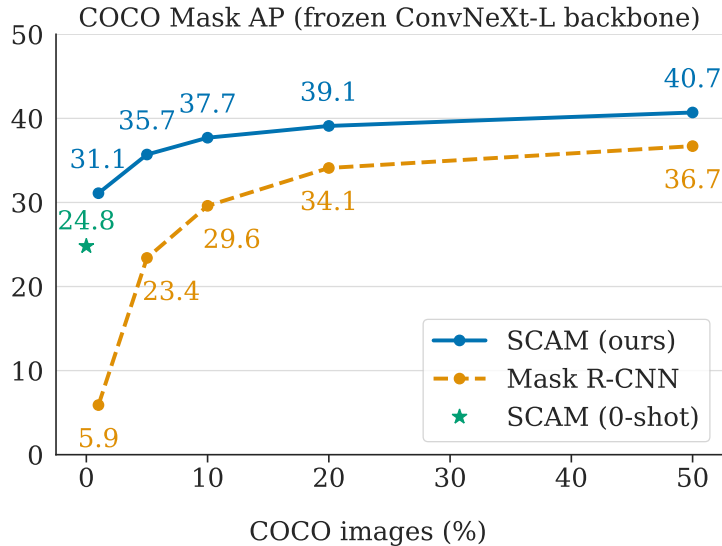


Figure 5.6: **Transferring SCAM using labeled masks.** When trained with the same backbone, SCAM outperforms Mask R-CNN, with more prominent gains in a low-data regime.

5.4.3 Transfer with Labeled Masks

Having shown results with unlabeled masks, we now aim to train SCAM with labeled masks. We train using the train2017 split and report Average Precision (AP_{mask}) on the val2017 split. To evaluate the data efficiency of SCAM, we also train using randomly sampled subsets of $\{1, 5, 10, 20, 50\}$ % COCO train2017 split.

In the spirit of using minimal computational resources, we keep the backbone and segmenter *completely frozen* and adopt a lightweight training schedule. We train for ≈ 16 COCO epochs². We only training the lightweight prompter and fine-tuning the classifier. These design choices make our setup accessible to a broad (academic) community – our models can be trained in 24 hours or less using only $2 \times$ A40 GPUs.

Baseline. We compare SCAM against Mask R-CNN [92], the standard region-based object detector in the existing literature. We train a Mask R-CNN with the same backbone as SAM and follow the same data restrictions. We ensure fair comparisons with our baseline by using the CLIP ConvNeXt-L as a frozen backbone, like in SCAM. We train Mask R-CNN for $2 \times$ longer as it is not equipped with pre-trained segmentation components, although we find that even longer schedules tend to overfit. We tune the optimization hyper-parameters for the Mask R-CNN baseline and use them for SCAM; see Appendix C for details.

²15K iterations, batch size = 64 for full COCO, iterations are scaled according to data subset size

Results. We present the performance of all methods at different amounts of data in Figure 5.6. We find that using CLIP with SAM in SCAM shows substantial gains as compared to using CLIP as a backbone in Mask R-CNN, especially when the training data is scarce. We also find that reverting to our zero-shot setup is sometimes beneficial if the amount of data is too scarce or there is no compute budget to train models on domain-specific data.

5.5 Conclusion

We show that SCAM can effectively consolidate the independent research efforts that build large-scale models for general image classification and image segmentation (but not both). Our extensive experiments on zero-shot transfer, followed by the adaptation of SCAM to unlabeled mask data show the flexibility of our design. Furthermore, in settings with scarce labeled mask data, we show that our architecture can outperform the current state-of-the-art architectures under tight resource constraints.

Chapter 6

Conclusion

In this dissertation, we explored the possibilities and potential advantages of using natural language supervision for computer vision. The research presented in the previous four chapters has offered compelling solutions for the four problems we initially laid out:

1. **Learning representations:** Using language supervision, **VirTex** (Chapter 2) provides a strong alternative to the deeply entrenched approach of learning visual representations using labeled image datasets like ImageNet.
2. **Scaling data:** We scale up VirTex models using **RedCaps** (Chapter 3), our web-curated dataset comprising 12 million image-text pairs, and show strong transfer results on downstream tasks involving image classification and image captioning.
3. **Understanding data:** Our representation learning method, **MERU** (Chapter 4), learns a *visual-semantic* hierarchy from image-text datasets like RedCaps, offering greater insight into the diversity of concepts underlying millions of images and text.
4. **Transfer to downstream tasks:** Our modular design of **SCAM** (Chapter 5) allows transferring pre-trained vision models for object detection and segmentation with minimal or no fine-tuning, offering practical benefits in data-starved settings.

Concurrent and follow-up studies

The idea of language-supervised visual representation learning has flourished in the past years, concurrent with this dissertation. ICMLM [23] was published concurrent to VirTex, which utilized language supervision for vision pre-training based on the masked language modeling objective [53]. These works were soon followed up by CLIP [198] and ALIGN [114], that scaled up learning to millions of image-text pairs and showed remarkable *zero-shot* transfer capabilities to image classification and retrieval tasks. An appealing property of these models was their ability to perform classification and retrieval based on user-provided text prompts, unlike prior ImageNet-supervised models that were restricted to the pre-defined label ontologies.

These methods use a contrastive learning objective, different from the generative objective of VirTex (image captioning), showcasing easier scalability with noisy image-text data. Subsequent studies like BLIP [146] and CoCa [272] combine generative and contrastive learning objectives to train vision-language models. Surprisingly, CapPa [242] shows that VirTex-style image captioning models also scale effectively at billion-scale data while using modern Vision Transformers [60] as image encoders.

Since its release, the research community has used our RedCaps dataset for a myriad of applications like text-to-image generation (Make-a-Scene [73], StyleGAN-T [209]), text-guided semantic segmentation (GroupViT [263]), self-supervised learning (LG-SSL [64]), and building general-purpose multimodal models (Unified-IO [165]). Follow-up works after RedCaps have developed orders of magnitude larger datasets, notably the LAION datasets (400 million and 2 billion sets [210, 211]) and DataComp [72]. Apart from having image-text pairs, datasets like MMC4 [290] and OBELISC [136] are structured as long-form documents with interleaved images and text.

While image-text data is abundant on the internet, ensuring high quality remains an uphill climb given the noisy and unstructured nature of the internet data. Also, the curation of internet data would always involve navigating the tricky waters of copyright and potential ethical risks regarding consent, the occurrence of personally identifiable information, and undesirable biases like imbalance representation of demographic groups [19]. For example, the LAION datasets have disabled public access as of January 2024 after the discovery of child abuse images [235].

Looking forward: Opportunities and challenges

In future research, the exploration into multimodal representation learning and model development will only continue to expand, to other modalities beyond language, *e.g.*, audio and haptics (that allow *actions*). This dissertation is one step in this overarching direction, underscoring the transformative potential of vision-language learning for computer vision.

The field is witnessing rapid adoption into real-world applications. Industry products for multimodal *Generative AI* such as GPT-4V [184] and Gemini [234] are reaching millions of daily users worldwide. These technologies offer tantalizing previews of what is possible and the remarkable advancements that seemed almost utopian before 2021. While these systems have room for improvements in terms of robustness, alignment, and interpretability, the strides made so far are inspiring. They signal that we are on the cusp of unprecedented advancements. Indeed, the best is yet to come.

Appendix A

Web-curated Image-Text Data from Reddit

A.1 List of subreddits in RedCaps

RedCaps comprises data from 350 subreddits (see Section 3.2.1). All subreddits are listed below alphabetically with the number of instances in each subreddit.

r/abandoned	7.0K	r/abandonedporn	56.2K	r/absoluteunits	33.9K
r/airplants	8.6K	r/alltheanimals	1.3K	r/amateurphotography	14.0K
r/amateurroomporn	11.6K	r/animalporn	15.8K	r/antiques	17.0K
r/antkeeping	3.0K	r/ants	1.2K	r/aquariums	139K
r/architectureporn	14.1K	r/artefactporn	9.6K	r/astronomy	2.1K
r/astrophotography	24.6K	r/australiancattledog	21.4K	r/australianshepherd	12.5K
r/autumnporn	2.7K	r/averagebattlestations	5.6K	r/awweducational	5.9K
r/awwnverts	7.6K	r/axolotls	9.7K	r/backpacking	8.9K
r/backyardchickens	17.3K	r/baking	119K	r/ballpython	19.2K
r/barista	6.6K	r/bassfishing	6.5K	r/battlestations	58.4K
r/bbq	22.3K	r/beagle	21.4K	r/beardeddragons	55.1K
r/beekeeping	1.2K	r/beerandpizza	1.2K	r/beerporn	95.7K
r/beerwithaview	8.9K	r/beginnerwoodworking	8.7K	r/bengalcats	7.0K
r/bento	4.8K	r/bernesemountaindogs	6.4K	r/berries	805
r/bettafish	64.7K	r/bicycling	80.8K	r/bikecommuting	9.8K
r/birding	21.1K	r/birdphotography	1.3K	r/birdpics	29.1K
r/birds	1.2K	r/birdsofprey	2.2K	r/blackcats	84.1K
r/blacksmith	13.3K	r/bladesmith	9.1K	r/boatporn	2.8K
r/bonsai	18.1K	r/bookporn	4.5K	r/bookshelf	6.2K
r/bordercollie	21.9K	r/bostonterrier	28.2K	r/botanicalporn	13.4K
r/breadit	71.6K	r/breakfast	2.2K	r/breakfastfood	3.8K
r/bridgeporn	2.4K	r/brochet	3.2K	r/budgetfood	1.6K
r/budgies	1.3K	r/bulldogs	24.1K	r/burgers	10.7K
r/butterflies	4.5K	r/cabinporn	2.7K	r/cactus	36.5K

r/cakedecorating	14.0K	r/cakewin	4.8K	r/cameras	3.3K
r/camping	21.4K	r/campingandhiking	25.5K	r/carnivorousplants	1.3K
r/carpentry	4.1K	r/carporn	102K	r/cassetteculture	12.2K
r/castiron	33.6K	r/castles	7.0K	r/casualknitting	3.1K
r/catpictures	51.9K	r/cats	643K	r/ceramics	4.8K
r/chameleons	7.9K	r/charcuterie	3.0K	r/cheese	5.0K
r/cheesemaking	1.7K	r/chefit	1.6K	r/chefknives	8.7K
r/chickens	9.6K	r/chihuahua	36.2K	r/chinchilla	5.6K
r/chinesefood	1.8K	r/churchporn	2.1K	r/cider	2.4K
r/cityporn	56.9K	r/classiccars	14.4K	r/cockatiel	12.1K
r/cocktails	25.0K	r/coffeestations	1.5K	r/coins	45.0K
r/cookie decorating	3.7K	r/corgi	64.7K	r/cornsnakes	3.4K
r/cozyplaces	44.9K	r/crafts	44.0K	r/crestedgecko	5.2K
r/crochet	125K	r/crossstitch	63.6K	r/crows	1.1K
r/crystals	24.0K	r/cupcakes	2.3K	r/dachshund	47.0K
r/damnthatsinteresting	28.4K	r/desertporn	1.2K	r/designmyroom	6.3K
r/desksetup	1.1K	r/dessert	3.2K	r/dessertporn	9.5K
r/diy	19.4K	r/dobermanpinscher	8.1K	r/doggos	18.6K
r/dogpictures	120K	r/drunkencookery	5.9K	r/duck	4.7K
r/dumpsterdiving	4.4K	r/earthporn	262K	r/eatsandwiches	20.5K
r/embroidery	38.5K	r/entomology	6.9K	r/equestrian	5.2K
r/espresso	8.5K	r/exposureporn	10.2K	r/eyebleach	80.9K
r/flporn	12.9K	r/farming	4.7K	r/femalelivingspace	947
r/fermentation	10.6K	r/ferrets	26.2K	r/fireporn	1.7K
r/fish	2.9K	r/fishing	51.0K	r/flowers	20.8K
r/flyfishing	19.1K	r/food	393K	r/foodporn	202K
r/foraging	9.5K	r/fossilporn	1.7K	r/fountainpens	52.8K
r/foxes	7.7K	r/frenchbulldogs	12.2K	r/frogs	14.8K
r/gardening	208K	r/gardenwild	1.0K	r/geckos	5.9K
r/gemstones	1.5K	r/geologyporn	1.9K	r/germanshepherds	46.0K
r/glutenfree	2.9K	r/gold	1.3K	r/goldenretrievers	42.4K
r/goldfish	3.9K	r/greatpyrenees	8.8K	r/grilledcheese	13.4K
r/grilling	12.6K	r/guineapigs	56.8K	r/gunporn	17.5K
r/guns	99.1K	r/hamsters	26.9K	r/handtools	3.2K
r/healthyfood	8.2K	r/hedgehog	1.7K	r/helicopters	3.2K
r/herpetology	9.7K	r/hiking	41.6K	r/homestead	9.3K
r/horses	16.3K	r/hotpeppers	27.8K	r/houseplants	182K
r/houseporn	2.8K	r/husky	35.9K	r/icecreamery	1.1K
r/indoorgarden	29.0K	r/infrastructureporn	7.0K	r/insects	20.4K
r/instantpot	2.8K	r/interestingasfuck	73.7K	r/interiordesign	6.7K
r/itookapicture	327K	r/jellyfish	713	r/jewelry	3.5K

r/kayakfishing	4.8K	r/kayaking	9.9K	r/ketorecipes	22.3K
r/knifeporn	2.5K	r/knives	63.9K	r/labrador	25.1K
r/leathercraft	16.0K	r/leopardgeckos	9.0K	r/lizards	2.4K
r/lookatmydog	43.2K	r/macarons	5.3K	r/machineporn	6.2K
r/macroporn	14.8K	r/malelivingspace	17.1K	r/mead	12.4K
r/mealprepsunday	33.1K	r/mechanicalkeyboards	156K	r/mechanicalpencils	5.3K
r/melts	1.2K	r/metalworking	3.8K	r/microgreens	1.1K
r/microporn	1.8K	r/mildlyinteresting	731K	r/mineralporn	10.4K
r/monitors	2.2K	r/monstera	6.9K	r/mostbeautiful	25.5K
r/motorcycleporn	6.4K	r/muglife	4.1K	r/mushroomgrowers	13.4K
r/mushroomporn	4.7K	r/mushrooms	5.6K	r/mycology	83.6K
r/natureisfuckinlit	61.3K	r/natureporn	10.1K	r/nebelung	4.6K
r/orchids	26.4K	r/otters	2.6K	r/outdoors	30.2K
r/owls	3.6K	r/parrots	38.0K	r/pelletgrills	4.5K
r/pens	5.0K	r/perfectfit	19.7K	r/permaculture	1.3K
r/photocritique	51.5K	r/photographs	11.5K	r/pics	1.9M
r/pitbulls	88.5K	r/pizza	46.5K	r/plantbaseddiet	3.7K
r/plantedtank	44.4K	r/plants	42.9K	r/plantsandpots	3.0K
r/pomeranians	7.4K	r/pottery	9.6K	r/pourpainting	15.3K
r/proplifting	17.8K	r/pug	5.1K	r/pugs	40.2K
r/quilting	24.1K	r/rabbits	105K	r/ramen	10.9K
r/rarepuppies	150K	r/reeftank	29.5K	r/reptiles	33.1K
r/resincasting	3.7K	r/roomporn	13.9K	r/roses	3.2K
r/rottweiler	11.5K	r/ruralporn	9.0K	r/sailing	10.5K
r/salsasnob	2.9K	r/samoyeds	6.8K	r/savagegarden	14.9K
r/scotch	32.1K	r/seaporn	2.2K	r/seriouseats	8.8K
r/sewing	29.7K	r/sharks	3.0K	r/shiba	27.8K
r/shihtzu	8.9K	r/shrimptank	14.7K	r/siamesecats	9.6K
r/siberiancats	2.7K	r/silverbugs	26.1K	r/skyporn	36.1K
r/sloths	5.9K	r/smoking	38.3K	r/snails	6.9K
r/snakes	45.4K	r/sneakers	314K	r/sneks	17.4K
r/somethingimade	50.4K	r/soup	1.5K	r/sourdough	32.2K
r/sousvide	13.6K	r/spaceporn	16.3K	r/spicy	12.4K
r/spiderbro	16.1K	r/spiders	41.9K	r/squirrels	8.1K
r/steak	19.8K	r/streetphotography	10.1K	r/succulents	201K
r/superbowl	7.5K	r/supermodelcats	33.6K	r/sushi	13.4K
r/tacos	2.7K	r/tarantulas	15.0K	r/tastyfood	2.3K
r/tea	20.5K	r/teaporn	1.2K	r/tequila	2.9K
r/terrariums	7.3K	r/thedepthsbelow	2.5K	r/thriftstorehauls	91.4K
r/tinyanimalsonfingers	3.1K	r/tonightsdinner	25.7K	r/toolporn	2.1K
r/tools	21.7K	r/torties	11.0K	r/tortoise	5.6K
r/tractors	2.3K	r/trailrunning	7.3K	r/trains	14.2K
r/trucks	30.4K	r/turtle	9.1K	r/underwaterphotography	1.2K

r/upcycling	1.9K	r/urbanexploration	18.8K	r/urbanhell	8.1K
r/veganfoodporn	18.7K	r/veganrecipes	9.9K	r/vegetablegardening	12.1K
r/vegetarian	9.8K	r/villageporn	6.4K	r/vintage	4.4K
r/vintageaudio	12.7K	r/vinyl	41.7K	r/volumeeating	2.1K
r/watches	64.2K	r/waterporn	9.6K	r/weatherporn	1.8K
r/wewantplates	17.0K	r/wildernessbackpacking	3.1K	r/wildlifephotography	16.3K
r/wine	12.7K	r/winterporn	7.0K	r/woodcarving	6.3K
r/woodworking	112K	r/workbenches	2.8K	r/workspaces	1.5K
r/yarnaddicts	2.6K	r/zerowaste	7.7K		

A.2 Datasheet for RedCaps dataset

Datasheets for datasets introduced by Gebru et al. [76] serve as a medium of communication between the creators and consumers (users) of a dataset. They effectively consolidate the motivation, creation process, composition, and intended uses of a dataset as a series of questions and answers. This appendix chapter provides a datasheet for the RedCaps dataset. It accompanies version v1.0 released in October 2021 with our accepted paper at the *NeurIPS 2021 Track on Datasets and Benchmarks* [51]. In this section:

- All mentions of *RedCaps* and all reported data statistics refer to RedCaps v1.0
- All mentions of *dataset website* refer to redcaps.xyz
- All mentions of *data collection code* refer to the `redcaps-downloader` repository available at github.com/redcaps-dataset/redcaps-downloader

Motivation

Q1. **For what purpose was the dataset created?** *Was there a specific task in mind? Was there a specific gap that needed to be filled? Please provide a description.*

- Large datasets of image-text pairs are widely used for pre-training generic representations that transfer to a variety of downstream vision and vision-language tasks. Existing public datasets of this kind were curated from search engine results (SBU Captions [185]) or HTML alt-text from arbitrary web pages (Conceptual Captions [30, 215]). They performed complex data filtering to deal with noisy web data. Due to aggressive filtering, their data collection is inefficient and diversity is artificially suppressed. We argue that the quality of data depends on its *source*, and the *human intent* behind its creation. To this end, we explore Reddit as a potential source for curating high-quality data. We introduce RedCaps – a large dataset of 12M image-text pairs from Reddit. While we expect the use-cases of RedCaps to be similar to

existing datasets, we discuss how Reddit as a data source leads to fast and lightweight collection, better data quality, lets us easily steer the data distribution, and facilitates ethically responsible data curation.

Q2. Who created this dataset (e.g., which team, research group) and on behalf of which entity (e.g., company, institution, organization)?

- Four researchers at the University of Michigan (affiliated as of 2021) have created RedCaps: Karan Desai, Gaurav Kaul, Zubin Aysola, and Justin Johnson.

Q3. Who funded the creation of the dataset? *If there is an associated grant, please provide the name of the grantor and the grant name and number.*

- We collected RedCaps without any monetary costs since no part of our dataset requires annotations from crowd workers or contractors. This research work was partially supported by the Toyota Research Institute (TRI). However, note that this article solely reflects the opinions and conclusions of its authors and not TRI or any other Toyota entity.

Q4. Any other comments?

- No.

Composition

Q5. What do the instances that comprise the dataset represent (e.g., documents, photos, people, countries)? *Are there multiple types of instances (e.g., movies, users, and ratings; people and interactions between them; nodes and edges)?*

- Each instance in RedCaps represents a single Reddit image post.

Q6. How many instances are there in total (of each type, if appropriate)?

- There are nearly 12M (12,011,111) instances in RedCaps.

Q7. Does the dataset contain all possible instances or is it a sample (not necessarily random) of instances from a larger set? *If the dataset is a sample, then what is the larger set? Is the sample representative of the larger set (e.g., geographic coverage)? If so, please describe how this representativeness was validated/verified. If it is not representative of the larger set, please describe why not (e.g., to cover a more diverse range of instances, because instances were withheld or unavailable).*

- RedCaps is a small sample drawn from all the data uploaded to Reddit. Millions of Reddit users submit image posts across thousands of subreddits on a daily basis. We hand-picked 350 subreddits containing high-quality photographs with descriptive

captions, while leaving out lots of subreddits focused on many other topics like politics, religion, science, and memes. Even within the selected subreddits, we filtered instances to improve data quality and mitigate privacy risks for people appearing in images. Hence, RedCaps data does not fully represent Reddit.

Q8. What data does each instance consist of? “Raw” data (e.g., unprocessed text or images) or features? *In either case, please provide a description.*

- Each instance in RedCaps consists of nine metadata fields:
 - "image_id": Unique alphanumeric ID of the image post (assigned by Reddit).
 - "author": Reddit username of the image post author.
 - "url": Static URL for downloading the image associated with the post.
 - "raw_caption": Textual description of the image, written by the post author.
 - "caption": Cleaned version of "raw_caption" by us (see [Q35](#)).
 - "subreddit": Name of subreddit where the post was submitted.
 - "score": Net upvotes (discounting downvotes) received by the image post.
 - "created_utc": Integer time epoch (in UTC) when the post was submitted to Reddit.
 - "permalink": Partial URL of the Reddit post (reddit.com/<permalink>).

Q9. Is there a label or target associated with each instance?

- No, we do not define any label or target for the instances. Targets are task-dependent. RedCaps can be used for a variety of tasks such as image captioning (*inputs = images, targets = captions*), image classification (*inputs = images, targets = subreddits*), text-to-image generation (*inputs = captions, targets = images*), or self-supervised visual learning (*inputs = images, no targets*).

Q10. Is any information missing from individual instances? *If so, please provide a description, explaining why this information is missing. This does not include intentionally removed information but might include, e.g., redacted text.*

- No, and yes. No, because all the metadata fields for every instance are filled with valid values. Yes, because the "url" for some instances may not retrieve the underlying image. This may happen if the Reddit user (author) removes the post from Reddit. Such deletions reduce our dataset size over time, however, post deletions are very rare after six months of creation.

Q11. Are relationships between individual instances made explicit (e.g., users' movie ratings, social network links)? *If so, please describe how they are made explicit.*

- Some implicit relationships do exist in our data. All instances belonging to the same subreddit are likely to have highly related visual and textual content. Moreover,

multiple images posted by a single Reddit user may be highly related (photos of their pets, cars, etc.).

Q12. Are there recommended data splits (e.g., training, development/validation, testing)? *If so, please provide a description of these splits, explaining the rationale behind them.*

- We intend our dataset to be primarily used for pre-training with one or more specific downstream task(s) in mind. Hence, all instances in our dataset would be used for training while the validation split is derived from the downstream task(s). If users require a validation split, we recommend sampling it such that it follows the same subreddit distribution as entire dataset.

Q13. Are there any errors, sources of noise, or redundancies in the dataset?

- RedCaps is noisy *by design* since image-text pairs on the internet are noisy and unstructured. Some instances may also have duplicate images and captions – Reddit users may have shared the same image post in multiple subreddits. Such redundancies constitute a very small fraction of the dataset and should have almost no effect in training large-scale models.

Q14. Is the dataset self-contained, or does it link to or otherwise rely on external resources (e.g., websites, tweets, other datasets)? *If it links to or relies on external resources, then –*

- (a) *Are there guarantees that they will exist, and remain constant, over time?*
- (b) *Are there official archival versions of the complete dataset (i.e., including the external resources as they existed at the time the dataset was created)?*
- (c) *Are there any restrictions (e.g., licenses, fees) associated with any of the external resources that might apply to a future user? Please provide descriptions of all external resources and any restrictions associated with them, as well as links or other access points, as appropriate.*
- We do not distribute images of our dataset to respect Reddit user privacy and to limit our storage budget. Instead, we provide image URLs (‘‘url’’, Q8) that point to images hosted on either Reddit, Imgur, or Flickr image servers. In response to sub-questions:
 - (a) These image servers ensure stable access unless the Reddit user deletes their image post.
 - (b) Yes, Reddit archives all the metadata of submitted posts. For images, Reddit only archives the URL and not the media content, giving full control of accessibility to the users.

(c) All image URLs are freely accessible. It is unlikely for the image servers to restrict access in the future, given their free accessibility over the past decade.

Q15. Does the dataset contain data that might be considered confidential (e.g., data that is protected by legal privilege or by doctor-patient confidentiality, data that includes the content of individuals non-public communications)?

- No, the subreddits included in RedCaps do not cover topics that may be considered confidential. All posts were publicly shared on Reddit prior to inclusion in RedCaps.

Q16. Does the dataset contain data that, if viewed directly, might be offensive, insulting, threatening, or might otherwise cause anxiety?

- The scale of RedCaps means that we are unable to verify the contents of all images and captions. However, we have tried to minimize the possibility that RedCaps contains data that might be offensive, insulting, threatening, or might cause anxiety. Refer Section 3.2.2 for details, our mitigations are as follows:
 - (a) We manually curate the set of subreddits from which to collect data; we only chose subreddits that are not marked NSFW and which generally contain non-offensive content.
 - (b) Within our curated subreddits, we did not include any posts marked NSFW.
 - (c) We removed all instances whose captions contained any of the 400 potentially offensive words or phrases¹.
 - (d) We remove all instances whose images were flagged NSFW by an off-the-shelf detector. We manually checked 50K random images in RedCaps and found one image containing nudity (exposed buttocks; no identifiable face).

Q17. Does the dataset relate to people?

- The dataset pertains to people in that people wrote the captions and posted images to Reddit that we curate in RedCaps. We made specific design choices while curating RedCaps to avoid large quantities of images containing people (refer Section 3.2.2):
 - (a) We collect data from manually curated subreddits in which most contain primarily pertains to animals, objects, places, or activities. We exclude all subreddits whose primary purpose is to share and describe images of people (such as celebrity photos or user selfies).
 - (b) We use an off-the-shelf face detector to find and remove images with potential presence of human faces. We manually checked 50K random images in RedCaps (Q16) and found 79 images with identifiable human faces – the entire dataset may have $\approx 19\text{K}$ (0.15%) images with identifiable people.

¹github.com/LDNOOBW/List-of-Dirty-Naughty-Obscene-and-Otherwise-Bad-Words

Q18. Does the dataset identify any subpopulations (e.g., by age, gender)? *If so, please describe how these subpopulations are identified and provide a description of their respective distributions within the dataset.*

- RedCaps does not explicitly identify any subpopulations. Since some images contain people and captions are free-form natural language written by Reddit users, it is possible that some captions may identify people appearing in individual images as part of a subpopulation.

Q19. Is it possible to identify one or more natural persons, either directly or indirectly (i.e., in combination with other data) from the dataset?

- Yes, all instances in RedCaps include Reddit usernames of their post authors. This could be used to look up the Reddit user profile, and some Reddit users may have identifying information in their profiles. Some images may contain human faces (Q17) which could be identified by appearance. However, note that all this information is already public on Reddit, and searching it in RedCaps is no easier than searching directly on Reddit.

Q20. Does the dataset contain data that might be considered sensitive in any way (e.g., data that reveals racial or ethnic origins, sexual orientations, religious beliefs, political opinions or union memberships, or locations; financial or health data; biometric or genetic data; forms of government identification, such as social security numbers; criminal history)?

- Highly unlikely, data from our manually selected subreddits does not contain sensitive information of the above forms. In case some instances have such information, then note that all this information is already publicly available on Reddit.

Q21. Any other comments?

- No.

Collection Process

Q22. How was the data associated with each instance acquired? *Was the data directly observable (e.g., raw text, movie ratings), reported by subjects (e.g., survey responses), or indirectly inferred/derived from other data (e.g., part-of-speech tags, model-based guesses for age or language)? If data was reported by subjects or indirectly inferred/derived from other data, was the data validated/verified? If so, please describe how.*

- We collected instance IDs using Pushshift API (pushshift.io) and the remaining metadata fields (Q8) using the Reddit API (reddit.com/wiki/api). All fields except

"caption" are available in API responses; "caption" is derived by applying text pre-processing to "raw_caption" field (Q35).

Q23. What mechanisms or procedures were used to collect the data (e.g., hardware apparatus or sensor, manual human curation, software program, software API)?

How were these mechanisms or procedures validated?

- We collected all data using resources at the University of Michigan. The code for querying APIs and filtering data is implemented in Python. We validated our implementation by manually checking a few RedCaps instances with their posts on [reddit.com](https://www.reddit.com).

Q24. If the dataset is a sample from a larger set, what was the sampling strategy?

- RedCaps is a small sample containing data from 350 subreddits out of thousands of subreddits on Reddit. We hand-picked each subreddit for our dataset based on its content. See Q7, Q16, and Q17 for details on how we selected each subreddit.

Q25. Who was involved in data collection process (e.g., students, crowd-workers, contractors) and how were they compensated (e.g., how much were crowd-workers paid)?

- Our data collection pipeline is fully automatic and does not require any human annotators. Reddit users have uploaded image posts whose metadata is a part of RedCaps – we did not directly interact with these users.

Q26. Over what timeframe was the data collected? Does this timeframe match the creation timeframe of the data associated with the instances (e.g., recent crawl of old news articles)? If not, please provide a description of the timeframe.

- RedCaps contains image posts that were uploaded to Reddit between 2008–2020. We collected all data in early 2021, which we used to conduct experiments for our NeurIPS 2021 submission. Since Reddit posts may get deleted over time, we exactly re-collected a fresh version in August 2021 after acceptance (and re-trained all our experiments). Reddit posts observe the most user activity (upvotes, comments, moderation) for six months after their creation – posts from 2008–2020 are less likely to be updated after August 2021.

Q27. Were any ethical review processes conducted (e.g., by an institutional review board)? If so, please provide a description of these review processes, including the outcomes, as well as a link or other access point to any supporting documentation.

- We did not conduct a formal ethical review process via institutional review boards. However, as described in Section 3.2.2 and Q16 we employed several filtering mech-

anisms to try and remove instances that could be problematic.

Q28. Does the dataset relate to people?

- Some images of RedCaps may contain images of people (see Q17).

Q29. Did you collect the data from the individuals in question directly, or obtain it via third parties or other sources (e.g., websites)?

- We collected data submitted by Reddit users indirectly through the Reddit API. However, users agree with Reddit’s User Agreement regarding redistribution of their data by Reddit.

Q30. Were the individuals in question notified about the data collection? If so, please describe (or show with screenshots or other information) how notice was provided, and provide a link or other access point to, or otherwise reproduce, the exact language of the notification itself.

- No. Reddit users are anonymous by default and are not required to share their personal contact information (email, phone numbers, etc.). Hence, the only way to notify the authors of RedCaps image posts is by sending them private messages on Reddit. This is practically difficult to do manually and will be classified as spam and blocked by Reddit if attempted to programmatically send a templated message to millions of users.

Q31. Did the individuals in question consent to the collection and use of their data? If so, please describe (or show with screenshots or other information) how consent was requested and provided, and provide a link or other access point to, or otherwise reproduce, the exact language to which the individuals consented.

- Users did not explicitly consent to the use of their data in our dataset. However, by uploading their data on Reddit, they consent that it would appear on the Reddit platform and will be accessible via the official Reddit API (which we use to collect RedCaps).

Q32. If consent was obtained, were the consenting individuals provided with a mechanism to revoke their consent in the future or for certain uses? If so, please provide a description, as well as a link or other access point to the mechanism (if appropriate).

- Users have full control over the presence of their data in our dataset. If users wish to revoke their consent, they can delete the underlying Reddit post – it will be automatically removed from RedCaps since we distributed images as URLs. Moreover, we provide an opt-out request form on our dataset website for anybody to request removal of an individual instance if it is potentially harmful (e.g. NSFW, violates

privacy, harmful stereotypes, etc.).

Q33. **Has an analysis of the potential impact of the dataset and its use on data subjects (e.g., a data protection impact analysis) been conducted?** *If so, please provide a description of this analysis, including the outcomes, as well as a link or other access point to any supporting documentation.*

– No.

Q34. **Any other comments?**

– No.

Preprocessing, Cleaning, and/or Labeling

Q35. **Was any preprocessing/cleaning/labeling of the data done (e.g., discretization or bucketing, tokenization, part-of-speech tagging, SIFT feature extraction, removal of instances, processing of missing values)?** *If so, please provide a description. If not, you may skip the remainder of the questions in this section.*

– We filtered all image posts with < 2 net upvotes, and those marked NSFW on Reddit. We remove character accents, emojis, non-latin characters, sub-strings enclosed in brackets (`(.*)`, `[.*]`), and replace social media handles (words starting with '@') with a special `[USR]` token. Refer Section 3.2.1 for main details. We also remove additional instances with focus on ethical considerations, see Q16, Q17 for more details.

Q36. **Was the “raw” data saved in addition to the preprocessed/cleaned/labeled data (e.g., to support unanticipated future uses)?** *If so, please provide a link or other access point to the “raw” data.*

– We provide the unprocessed captions obtained as-is from Reddit as part of our annotations (see "raw_caption" in Q8). However, we entirely discard all instances that were filtered with ethical considerations – based on the presence of faces, NSFW content, or harmful language.

Q37. **Is the software used to preprocess/clean/label the instances available?** *If so, please provide a link or other access point.*

– Yes, the data collection code is open-sourced; accessible from the dataset website.

Q38. **Any other comments?**

– No.

Uses

Q39. Has the dataset been used for any tasks already?

- We have used our dataset to train neural networks that perform image captioning, and that learn transferable visual representations for a variety of downstream visual recognition tasks (image classification, object detection, instance segmentation).

Q40. Is there a repository that links to any or all papers or systems that use the dataset? If so, please provide a link or other access point.

- We do not maintain such a repository. However, citation trackers like Google Scholar and Semantic Scholar would list all future works that cite our dataset.

Q41. What (other) tasks could the dataset be used for?

- We anticipate that the dataset could be used for a variety of vision-language tasks, such as image or text retrieval or text-to-image synthesis.

Q42. Is there anything about the composition of the dataset or the way it was collected and preprocessed/cleaned/labeled that might impact future uses? For example, is there anything that a future user might need to know to avoid uses that could result in unfair treatment of individuals or groups (e.g., stereotyping, quality of service issues) or other undesirable harms (e.g., financial harms, legal risks) Is there anything a future user could do to mitigate these undesirable harms?

- This is very difficult to anticipate. Future users of our dataset should be aware of Reddit’s user demographics which might subtly influence the types of images, languages, and ideas that are present in the dataset (refer Section 3.2.2). Moreover, users should be aware that our dataset intentionally excludes data from subreddits whose primary purpose is to share images that depict or describe people.

Q43. Are there any tasks for which the dataset should not be used?

- Broadly speaking, our dataset should only be used for non-commercial academic research. Our dataset should not be used for any tasks that involve identifying features related to people (facial recognition, gender, age, ethnicity identification, etc.) or make decisions that impact people (mortgages, job applications, criminal sentences; or moderation decisions about user-uploaded data that could result in bans from a website). Any commercial and for-profit uses of our dataset are restricted – it should not be used to train models that will be deployed in production systems as part of a product offered by businesses or government agencies.

Q44. Any other comments?

- No.

Distribution

Q45. Will the dataset be distributed to third parties outside of the entity (e.g., company, institution) on behalf of which the dataset was created?

- Yes, our dataset will be publicly available.

Q46. How will the dataset will be distributed (e.g., tarball on website, API, GitHub)
Does the dataset have a digital object identifier (DOI)?

- We distribute our dataset as a ZIP file containing all the annotations (JSON files). Users will have to download the images by themselves by using our data collection code. All uses of RedCaps should cite the NeurIPS 2021 paper as the reference.

Q47. When will the dataset be distributed?

- The dataset will be publicly available starting from October 2021.

Q48. Will the dataset be distributed under a copyright or other intellectual property (IP) license, and/or under applicable terms of use (ToU)? *If so, please describe this license/ToU, and provide a link or other access point to, or otherwise reproduce, any relevant licensing terms or ToU, as well as any fees associated with these restrictions.*

- Uses of our dataset are subject to Reddit API terms ([reddit.com/wiki/api-terms](https://www.reddit.com/wiki/api-terms)). Additionally users must comply with Reddit User Agreement, Content Policy, and Privacy Policy – all accessible at [redditinc.com/policies](https://www.redditinc.com/policies). The data collection code is released with an MIT license.

Q49. Have any third parties imposed IP-based or other restrictions on the data associated with the instances? *If so, please describe these restrictions, and provide a link or other access point to, or otherwise reproduce, any relevant licensing terms, as well as any fees associated with these restrictions.*

- The images corresponding to our instances are legally owned by Reddit users. Our dataset users can download them from the URLs we provide in annotation files, but redistributing images for commercial use is prohibited.

Q50. Do any export controls or other regulatory restrictions apply to the dataset or to individual instances? *If so, please describe these restrictions, and provide a link or other access point to, or otherwise reproduce, any supporting documentation.*

- No.

Q51. Any other comments?

- No.

Maintenance

Q52. Who will be supporting/hosting/maintaining the dataset?

- The dataset is hosted using Dropbox service provided by the University of Michigan. All the information about the dataset, including links to the paper, code, and future announcements will be accessible at the dataset website (redcaps.xyz).

Q53. How can the owner/curator/manager of the dataset be contacted?

- The contact emails of authors is available on the dataset website.

Q54. Is there an erratum? *If so, please provide a link or other access point.*

- There is no erratum for our initial release. We will version all errata as future releases (Q55) and document them on the dataset website.

Q55. Will the dataset be updated (e.g., to correct labeling errors, add new instances, delete instances)? *If so, please describe how often, by whom, and how updates will be communicated to users (e.g., mailing list, GitHub)?*

- We will update our dataset once every year and announce it on the dataset website. These future versions would include new instances corresponding to image posts made in 2021 and beyond, would remove instances that were requested to be removed via the opt out form (Q32).

Q56. If the dataset relates to people, are there applicable limits on the retention of the data associated with the instances (e.g., were individuals in question told that their data would be retained for a fixed period of time and then deleted)? *If so, please describe these limits and explain how they will be enforced.*

- Some images in RedCaps may depict people (Q17). Rather than directly distributing images, we distribute URLs that point to the original images uploaded by Reddit users. This means that users retain full control of their data – any post deleted from Reddit will be automatically removed from RedCaps (see also Q10, Q14, Q31).

Q57. Will older versions of the dataset continue to be supported/hosted/maintained? *If so, please describe how. If not, please describe how its obsolescence will be communicated to users.*

- A new version release of RedCaps will automatically deprecate its previous version. We will only support and maintain the latest version at all times. Deprecated versions

will remain accessible on the dataset website for a few weeks, after which they will be removed. We decided to deprecate old versions to ensure that any data that is requested to be removed (Q32) will be no longer accessible in future versions.

Q58. If others want to extend/augment/build on/contribute to the dataset, is there a mechanism for them to do so? Will these contributions be verified? If so, please describe how. If not, why not? Is there a process for communicating/distributing these contributions to other users?

- Anyone can extend RedCaps by using our data collection code (linked on the website). We are open to accepting extensions via personal communication with contributors. Otherwise, our code and data licenses allow others to create independent derivative works (with proper attribution) as long as they are used for non-commercial academic research.

Appendix B

Hyperbolic Image-Text Representations

B.1 Entailment loss derivations

We derive the entailment loss components (Eqn. 4.12) used in our approach. Note that for $c > 0$, the curvature of the hyperboloid is $-c$.

Half-aperture. To derive the entailment loss for arbitrary curvatures $c > 0$, we start with the expression of half-aperture for the Poincaré ball, introduced by Ganea et al. [74]. Let \mathbf{x}_b be a point on the Poincaré ball, the cone half-aperture is defined as follows:

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

The Poincaré ball model and Lorentz hyperboloid model are isometric to each other – one can map any point \mathbf{x}_b from the Poincaré ball to another point \mathbf{x}_h on the hyperboloid using the following differentiable transformation:

$$\mathbf{x}_h = \frac{2\mathbf{x}_b}{1 - c \|\mathbf{x}_b\|^2} \quad (\text{B.2})$$

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. B.2 in Eqn. B.1, we get the expression:

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

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 [141]. Let the

Lorentzian distances (Eqn. 4.4) 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. Making both substitutions in the above equation, we get:

$$\text{ext}(\mathbf{x}, \mathbf{y}) = \cos^{-1} \left[\frac{g(x) - g(z)g(y)}{\sqrt{g(z)^2 - 1} \sqrt{g(y)^2 - 1}} \right] \quad (\text{B.3})$$

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

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

We have $g(x) = -c \langle \mathbf{O}, \mathbf{y} \rangle_{\mathcal{L}}$ and $g(y) = -c \langle \mathbf{O}, \mathbf{x} \rangle_{\mathcal{L}}$. The Lorentzian inner product (Eqn. 4.1) involving origin $\mathbf{O} = [0, \sqrt{1/c}]$ as one of the operands has a simplified form:

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

We obtain $g(x) = x_{time}\sqrt{c}$ and $g(y) = y_{time}\sqrt{c}$. Finally, we can substitute $g(x)$, $g(y)$, and $g(z)$ in Eqn. B.3, along with the relation between x_{time} and \mathbf{x}_{space} (Eqn. 4.3), to give the final expression of the exterior angle as follows:

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

B.2 MERU evaluation datasets

In Section 4.4, we evaluated our trained MERU and CLIP models using twenty image classification datasets. Their details are listed in Table B.1. We use two open-source libraries to access these datasets – tensorflow-datasets and torchvision¹. We require every dataset to have a *train*, *validation*, and *test* split. We closely follow prior works of Radford et al. [198] and Mu et al. [176] to define these splits for every dataset:

- If all three splits are defined officially, we use them without modification.
- If an official *test* split does not exist, we use the official *validation* split as test split.
- If an official *validation* split does not exist or it is used as a *test* split, then we hold out a random subset of *train* split as the *validation* split.

Two datasets (EuroSAT and RESISC) do not define any splits, for these, we randomly sample three non-overlapping splits. Note that CLEVR Counts is derived from CLEVR [116] and SST2 was introduced as an NLP dataset by Socher et al. [218].

Dataset	Classes	Train	Val	Test	Dataset	Classes	Train	Val	Test
ImageNet [47]*	1000	–	–	50000	Caltech-101 [69]	102	2448	612	6084
Food-101 [21]	101	68175	7575	25250	Flowers [181]	102	1020	1020	6149
CIFAR-10 [130]	10	45000	5000	10000	STL-10 [43]	10	4000	1000	8000
CIFAR-100 [130]	100	45000	5000	10000	<u>EuroSAT</u> [97]	10	5000	5000	5000
CUB-2011 [251]	200	4795	1199	5794	<u>RESISC</u> [39]	45	3150	3150	25200
SUN397 [259]	397	15880	3970	19849	Country211 [198]	211	31650	10550	21100
Stanford Cars [126]	196	6515	1629	8041	MNIST [140]	10	48000	12000	10000
FGVC Aircraft [167]	100	3334	3333	3333	CLEVR Counts [278]	8	4500	500	5000
DTD [41]	47	1880	1880	1880	PCAM [246]	2	262144	32768	32768
Oxf-IIIT Pets [186]	37	2944	736	3669	SST2 [198]	2	6920	872	1821

Table B.1: **Evaluation datasets for MERU and CLIP.** Datasets in highlighted rows do not have an official validation split – we use a random held-out subset of the training split. Underlined datasets do not define any splits; we randomly sample non-overlapping splits. *: *ImageNet is not used for linear probe evaluation so other splits are not necessary.*

Zero-shot evaluation (Section 4.4.3). We report top-1 mean per-class accuracy on the *test* split of every dataset. Since this evaluation does not require any additional training on downstream data, it does not require *train* and *validation* splits. Table B.2 contains prompt templates we used to create classifier weights for every dataset.

Linear probe evaluation (Appendix B.4). We train using *train* split and search for the best hyperparameters using the *validation* split. Finally, we train a classifier using combined *train* + *validation* splits and report top-1 mean per-class accuracy on the *test* split.

¹[tensorflow.org/datasets](https://www.tensorflow.org/datasets) and pytorch.org/vision

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)	DTD (our prompts)	Oxford Flowers (our prompts)
food : {}.	pics : {} texture.	flowers : {}.
food porn : {}.	pics : {} pattern.	STL10
CIFAR-10 and CIFAR-100	pics : {} thing.	a photo of a {}.
a photo of a {}.	pics : this {} texture.	a photo of the {}.
a blurry photo of a {}.	pics : this {} pattern.	EuroSAT
a black and white photo of a {}.	pics : this {} thing.	a centered satellite photo of {}.
a low contrast photo of a {}.	Oxford-IIIT Pets	a centered satellite photo of a {}.
a high contrast photo of a {}.	a photo of a {}, a type of pet.	a centered satellite photo of the {}.
a bad photo of a {}.	Caltech-101	RESISC
a good photo of a {}.	a photo of a {}.	satellite imagery of {}.
a photo of a small {}.	a painting of a {}.	aerial imagery of {}.
a photo of a big {}.	a plastic {}.	satellite photo of {}.
a photo of the {}.	a sculpture of a {}.	aerial photo of {}.
a blurry photo of the {}.	a sketch of a {}.	satellite view of {}.
a black and white photo of the {}.	a tattoo of a {}.	aerial view of {}.
a low contrast photo of the {}.	a toy {}.	satellite imagery of a {}.
a high contrast photo of the {}.	a rendition of a {}.	aerial imagery of a {}.
a bad photo of the {}.	a embroidered {}.	satellite photo of a {}.
a good photo of the {}.	a cartoon {}.	aerial photo of a {}.
a photo of the small {}.	a {} in a video game.	satellite view of a {}.
a photo of the big {}.	a plushie {}.	aerial view of a {}.
CUB-2011 (our prompts)	a origami {}.	satellite imagery of the {}.
bird pics : {}.	art of a {}.	aerial imagery of the {}.
birding : {}.	graffiti of a {}.	satellite photo of the {}.
birds : {}.	a drawing of a {}.	aerial photo of the {}.
bird photography : {}.	a doodle of a {}.	satellite view of the {}.
SUN397	a photo of the {}.	aerial view of the {}.
a photo of a {}.	a painting of the {}.	Country211
a photo of the {}.	the plastic {}.	a photo i took in {}.
Stanford Cars	a sculpture of the {}.	a photo i took while visiting {}.
a photo of a {}.	a sketch of the {}.	a photo from my home country of {}.
a photo of the {}.	a tattoo of the {}.	a photo from my visit to {}.
a photo of my {}.	the toy {}.	a photo showing the country of {}.
i love my {}!	a rendition of the {}.	MNIST
a photo of my dirty {}.	the embroidered {}.	a photo of the number: "{}".
a photo of my clean {}.	the cartoon {}.	CLEVR
a photo of my new {}.	the {} in a video game.	a photo of {} objects.
a photo of my old {}.	the plushie {}.	Patch Camelyon
FGVC Aircraft	the origami {}.	this is a photo of {}.
a photo of a {}, a type of aircraft.	art of the {}.	Rendered SST2
a photo of the {}, a type of aircraft.	graffiti of the {}.	a {} review of a movie.
	a drawing of the {}.	
	a doodle of the {}.	

Table B.2: Prompts used for zero-shot image classification. Most prompts are similar to Radford et al. [198] except for a few datasets on which we observed significant improvements for both MERU and CLIP using our custom prompts. 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.

B.3 Developing a strong CLIP baseline

One of our contributions is to establish a lightweight, yet strong CLIP baseline. The original CLIP models [198] 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 less than one day.

We start with a reference CLIP ViT-S/16 baseline from SLIP [176] and carefully introduce one modification at a time. We benchmark improvements on zero-shot image classification across 16 datasets used in our main experiments, using text prompts used by Radford et al. [198]. Results are shown in Table B.3.

CLIP baseline by SLIP. This re-implemented baseline was trained using a 15M subset of the YFCC dataset [236]. We re-evaluate the publicly released ViT-S/16 checkpoint² 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:

- 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.
- The above change allows using weight decay = 0.2 like OpenAI’s CLIP, unlike 0.5 used by the CLIP re-implementation of Mu et al. [176].
- We resize input images using *bicubic* interpolation like original CLIP instead of *bilinear* interpolation used in the CLIP re-implementation of Mu et al. [176].

Fitting the model on 8-GPUs. This CLIP model requires 16× V100 32GB GPUs with a batch size of 4096 and automatic mixed precision [172]. Techniques like gradient checkpointing [31] can reduce memory requirements, but it comes at the 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).

²github.com/facebookresearch/slip

	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																		
Mu et al. [176] CLIP	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
+ <i>sin-cos pos embed</i>	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

Table B.3: **CLIP baseline.** We develop a strong CLIP baseline that trains on an 8-GPU machine in less than one day (ViT-S image encoder), starting with SLIP [176] 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.

Training with RedCaps dataset. RedCaps dataset [51] comprises 12M image-text pairs from Reddit, sourced from Pushshift [14]. 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 [21] and Pets [186]. Since RedCaps is smaller, 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 (refer Table B.2). For example, many captions in r/food simply mention the name of the dish in the corresponding image, hence we use the prompt ‘food : {}’. 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.

B.4 Linear probe evaluation

Our goal is to learn hyperbolic representations with MERU that capture a visual-semantic hierarchy underlying image-text datasets. Our experimental evaluations focus on *zero-shot* transfer [65, 198]. Another established protocol to evaluate visual representations is *linear probe evaluation*, which involves training linear models on *frozen* image embeddings. This protocol is popular in self-supervised representation learning literature, with Doersch et al. [56], Zhang et al. [282], and Noroozi and Favaro [182] being notable early works. We

		Food-101	CIFAR-10	CIFAR-100	CUB	SUN397	Cars	Aircraft	DTD	Pets	Caltech-101	Flowers	STL-10	EuroSAT	RESISC45	Country211	MINIST	CLEVR	PCAM	SST2
VT S/16	CLIP	85.3	89.6	72.3	68.8	61.1	60.5	42.2	71.2	87.9	88.4	96.2	95.5	95.7	88.1	15.0	98.5	57.5	84.6	54.9
	MERU	85.2	89.7	70.9	69.2	59.6	58.0	43.1	70.2	87.5	85.6	95.5	95.5	95.8	87.0	14.8	98.2	56.8	84.1	54.5
VT B/16	CLIP	88.4	92.2	76.5	73.2	64.7	71.1	50.4	72.6	90.2	89.6	97.3	97.1	96.9	90.0	16.7	98.9	52.7	84.4	57.6
	MERU	88.2	92.3	74.6	70.9	63.4	68.4	48.2	70.7	90.3	88.6	96.6	96.7	96.5	89.0	16.5	98.7	56.0	85.5	56.2
VT L/16	CLIP	89.6	95.3	80.5	75.7	66.0	75.7	54.5	75.7	92.0	92.0	97.4	97.6	96.9	90.5	17.8	99.2	55.6	87.5	56.1
	MERU	89.0	94.1	77.3	74.2	63.7	71.9	51.2	70.9	90.1	87.5	96.7	97.3	96.8	89.1	17.0	98.9	55.4	86.0	55.8

Table B.4: **Linear probe evaluation.** We train a logistic regression classifier on embeddings extracted from the image encoders of CLIP and MERU (before projection layers). Note that embeddings from MERU are *not* lifted onto the hyperboloid.

follow the implementation of Kornblith et al. [125] as it is simple and less sensitive to the choice of evaluation hyperparameters. This setup is also followed by CLIP [198] and many recent works on representation learning [64, 71, 150].

We evaluate using datasets listed in Table B.1. We train a logistic regression classifier on embeddings extracted from the image encoder (before the projection layer) of MERU and CLIP. For MERU, these underlying representations belong to a Euclidean space. We use the implementation from `scikit-learn` [190] library, with L-BFGS [156] optimizer and search the regularization cost per dataset, $C \in [10^{-6}, 10^6]$, performing two-step search on *validation* split. Using the best hyperparameters, we train a final classifier on combined *train + validation* splits for a maximum of 1000 iterations, then report top-1 mean per-class accuracy on the *test* split.

Results in Table B.4 show that MERU mostly matches or underperforms CLIP. Our main focus is not on improving the underlying Euclidean representations from the encoders, but on showing the zero-shot transfer and interpretability benefits of MERU. Future work can focus on improving MERU’s capabilities on other transfer applications.

B.5 Image traversals with YFCC captions

Our image traversals (Section 4.5.3) involve inferring the learned visual-semantic hierarchy in the representation space. Here we include additional qualitative results using captions from the YFCC dataset [236] to create the retrieval pool. We use *text descriptions* of a popular subset of this dataset – YFCC-15M [198]. We minimally process the text according to the RedCaps dataset to match the training data distribution. This involves converting to lowercase, using `ftfy` [222] to strips accents and non-latin characters, removing all sub-strings enclosed in brackets (`(.*)`, `[.*]`), and replacing social media handles (words

starting with ‘@’) with a <usr>. We also remove captions having more than 20 tokens (for ease of visualization). Finally, we obtain ≈ 8.7 M captions.

Image credits. Images displayed in Section 4.5.3 are collected from [pexels.com](https://www.pexels.com), a photography website that offers images with permissible usage licenses. Below is the list of the image source URLs listed in order of their appearance in Chapter 4. We thank all the photographers for generously sharing these images.

Illustration of the visual-semantic hierarchy (Figure 4.1).

- [pexels.com/photo/adult-yellow-labrador-retriever-standing-on-snow-field-1696589](https://www.pexels.com/photo/adult-yellow-labrador-retriever-standing-on-snow-field-1696589)
- [pexels.com/photo/homeless-cat-fighting-with-dog-on-street-6601811](https://www.pexels.com/photo/homeless-cat-fighting-with-dog-on-street-6601811)
- [pexels.com/photo/short-coated-gray-cat-20787](https://www.pexels.com/photo/short-coated-gray-cat-20787)

Image traversals – selected results (Figure 4.6).

- (1) [pexels.com/photo/a-bengal-cat-sitting-beside-wheatgrass-on-a-white-surface-7123957](https://www.pexels.com/photo/a-bengal-cat-sitting-beside-wheatgrass-on-a-white-surface-7123957)
- (2) [pexels.com/photo/white-horse-running-on-green-field-1996337](https://www.pexels.com/photo/white-horse-running-on-green-field-1996337)
- (3) [pexels.com/photo/photography-of-rainbow-during-cloudy-sky-757239](https://www.pexels.com/photo/photography-of-rainbow-during-cloudy-sky-757239)
- (4) [pexels.com/photo/retro-photo-camera-on-table-7162551](https://www.pexels.com/photo/retro-photo-camera-on-table-7162551)
- (5) [pexels.com/photo/avocado-toast-served-on-white-plate-10464867](https://www.pexels.com/photo/avocado-toast-served-on-white-plate-10464867)
- (6) [pexels.com/photo/photo-of-brooklyn-bridge-new-york-2260783](https://www.pexels.com/photo/photo-of-brooklyn-bridge-new-york-2260783)
- (7) [pexels.com/photo/taj-mahal-through-an-arch-2413613](https://www.pexels.com/photo/taj-mahal-through-an-arch-2413613)
- (8) [pexels.com/photo/sydney-opera-house-7088958](https://www.pexels.com/photo/sydney-opera-house-7088958)

Image traversals – locations and landmarks (Figure 4.7).

- (9) [pexels.com/photo/golden-gate-bridge-san-francisco-california-1141853](https://www.pexels.com/photo/golden-gate-bridge-san-francisco-california-1141853)
- (10) [pexels.com/photo/white-cliffs-of-dover-in-england-9692909](https://www.pexels.com/photo/white-cliffs-of-dover-in-england-9692909)
- (11) [pexels.com/photo/the-famous-fountain-paint-pots-in-yellowstone-national-park-12767016](https://www.pexels.com/photo/the-famous-fountain-paint-pots-in-yellowstone-national-park-12767016)
- (12) [pexels.com/photo/the-parthenon-temple-ruins-in-athens-greece-14446783](https://www.pexels.com/photo/the-parthenon-temple-ruins-in-athens-greece-14446783)
- (13) [pexels.com/photo/famous-big-ben-under-cloudy-sky-14434677](https://www.pexels.com/photo/famous-big-ben-under-cloudy-sky-14434677)
- (14) [pexels.com/photo/karlskirche-church-7018621](https://www.pexels.com/photo/karlskirche-church-7018621)
- (15) [pexels.com/photo/mt-fuji-3408353](https://www.pexels.com/photo/mt-fuji-3408353)
- (16) [pexels.com/photo/horseshoe-bend-arizona-2563733](https://www.pexels.com/photo/horseshoe-bend-arizona-2563733)
- (17) [pexels.com/photo/stars-at-night-1906667](https://www.pexels.com/photo/stars-at-night-1906667)
- (18) [pexels.com/photo/volcano-erupting-at-night-under-starry-sky-4220967](https://www.pexels.com/photo/volcano-erupting-at-night-under-starry-sky-4220967)
- (19) [pexels.com/photo/northern-lights-1933319](https://www.pexels.com/photo/northern-lights-1933319)
- (20) [pexels.com/photo/attraction-building-city-hotel-415999](https://www.pexels.com/photo/attraction-building-city-hotel-415999)

Image traversals – flora and fauna (Figure 4.8).

- (21) [pexels.com/photo/squirrel-up-on-the-snow-covered-tree-15306429](https://www.pexels.com/photo/squirrel-up-on-the-snow-covered-tree-15306429)
- (22) [pexels.com/photo/a-seagull-flying-under-blue-sky-12509256](https://www.pexels.com/photo/a-seagull-flying-under-blue-sky-12509256)
- (23) [pexels.com/photo/cute-pug-sitting-on-floor-in-white-kitchen-11199295](https://www.pexels.com/photo/cute-pug-sitting-on-floor-in-white-kitchen-11199295)
- (24) [pexels.com/photo/three-zebras-2118645](https://www.pexels.com/photo/three-zebras-2118645)
- (25) [pexels.com/photo/monarch-butterfly-perching-on-red-flower-1557208](https://www.pexels.com/photo/monarch-butterfly-perching-on-red-flower-1557208)
- (26) [pexels.com/photo/red-hibiscus-in-bloom-5801054](https://www.pexels.com/photo/red-hibiscus-in-bloom-5801054)
- (27) [pexels.com/photo/white-chicken-on-green-grass-field-58902](https://www.pexels.com/photo/white-chicken-on-green-grass-field-58902)

- (28) pexels.com/photo/yellow-blue-and-white-macaw-perched-on-brown-tree-branch-12715261
- (29) pexels.com/photo/closeup-photo-of-red-and-white-mushroom-757292
- (30) pexels.com/photo/photo-of-jellyfish-lot-underwater-3616240
- (31) pexels.com/photo/yellow-labrador-retriever-wearing-red-cap-4588002
- (32) pexels.com/photo/an-orca-whale-jumping-out-of-the-water-7767974

Image traversals – food and drinks (Figure 4.9).

- (33) pexels.com/photo/bread-and-coffee-for-breakfast-15891938
- (34) pexels.com/photo/grilled-cheese-on-a-plate-14941252
- (35) pexels.com/photo/bowl-of-ramen-12984979
- (36) pexels.com/photo/green-chili-peppers-and-a-knife-5792428
- (37) pexels.com/photo/spinach-caprese-salad-on-white-ceramic-plate-4768996
- (38) pexels.com/photo/chocolate-cupcakes-635409
- (39) pexels.com/photo/pav-bhaji-dish-on-a-bowl-5410400
- (40) pexels.com/photo/clear-glass-bottle-filled-with-broccoli-shake-1346347
- (41) pexels.com/photo/vada-pav-15017417
- (42) pexels.com/photo/old-fashioned-cocktail-drink-4762719
- (43) pexels.com/photo/coffee-in-white-ceramic-teacup-on-white-ceramic-suacer-894696
- (44) pexels.com/photo/espresso-martini-in-close-up-photography-15082368

Image traversals – objects and scenes (Figure 4.10).

- (45) pexels.com/photo/photograph-of-a-burning-fire-672636
- (46) pexels.com/photo/white-clouds-in-blue-sky-8354530
- (47) pexels.com/photo/raining-in-the-city-2448749
- (48) pexels.com/photo/aerial-view-of-road-in-the-middle-of-trees-1173777
- (49) pexels.com/photo/mountain-bike-on-the-beach-10542237
- (50) pexels.com/photo/wax-candles-burning-on-ground-14184952
- (51) pexels.com/photo/white-wooden-shelf-beside-bed-2062431
- (52) pexels.com/photo/stainless-steel-faucet-on-white-ceramic-sink-3761560
- (53) pexels.com/photo/jack-o-lantern-with-light-5659699
- (54) pexels.com/photo/black-and-white-piano-keys-4077310
- (55) pexels.com/photo/assorted-gift-boxes-on-floor-near-christmas-tree-3394779
- (56) pexels.com/photo/garden-table-and-chair-14831985

Image traversals – objects and scenes (Figure 4.11).

- (57) pexels.com/photo/turned-on-floor-lamp-near-sofa-on-a-library-room-1907784
- (58) pexels.com/photo/ripe-pineapple-on-gray-rock-beside-body-of-water-29555
- (59) pexels.com/photo/close-up-shot-of-a-cockatiel-13511241
- (60) pexels.com/photo/antique-bills-business-cash-210600



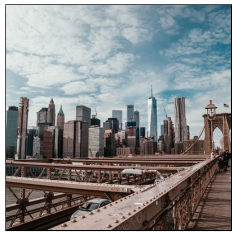
(1)

MERU	CLIP
<i>leopard and stig have a beautiful piano at their home.</i>	<i>loki is a 1 year old bengal cat.</i>
<i>merlin wasn't impressed to leave the last house and his precious cat grass</i>	↓
<i>my parents cat 'barry' loves being photographed!</i>	↓
<i>house cat posing</i>	↓
<i>mr . bo-majed</i>	↓
<i>our cat, our love our third member of our family. :)</i>	↓
<i>why are you taking pictures? it's dilo don't fill it up. :)</i>	↓
[ROOT]	[ROOT]



(2)

MERU	CLIP
<i>caught my attention by the beautiful light cascading on a grass behind this fellow.</i>	<i>pity about the camera shake in the evening light</i>
<i>the focus is all wrong , but the white on the tail and the tongue are pretty cool.</i>	↓
<i>just a goofy white guy.</i>	↓
<i>he was an active one, running to and fro.</i>	↓
<i>but then, she was happy to pose for me</i>	↓
<i>if she were a race horse her name would be poopbiscuit.</i>	↓
<i>dorky photo is dorky</i>	↓
<i>she looks so leery of the camera in this photo.</i>	↓
<i>this is only luky.</i>	↓
[ROOT]	[ROOT]



(3)

MERU	CLIP
<i>going across brooklyn bridge on the way to brooklyn 3 likes on instagram</i>	<i>shot from the manhattan end of the brooklyn bridge</i>
<i>manhattan depuis le brooklyn bridge park, a brooklyn.</i>	<i>much more scenic to walk on than the brooklyn bridge</i>
<i>bridge, manhattan skyline</i>	<i>new york new york!</i>
<i>shot from near the middle of the brooklyn bridge.</i>	↓
<i>this city goes on forever</i>	↓
<i>the city that never sleeps</i>	↓
<i>the city that never sleeps...it can't.</i>	↓
<i>it can be seen from most places in the city</i>	↓
[ROOT]	[ROOT]

Figure B.1: Image traversals (1/20) using a set of captions from the YFCC dataset.



(4)

MERU	CLIP
<i>avocado, roasted garlic and sriracha on light rye & raisins sourdough.</i>	<i>la tartine</i>
<i>with avocado slices - yum!</i>	<i>a toast to the new place</i>
<i>vaguely-healthy</i>	↓
<i>on bread, probably not what you should do with it, but it was a good meal</i>	↓
<i>nice if you like this sort of thing.</i>	↓
[ROOT]	[ROOT]



(5)

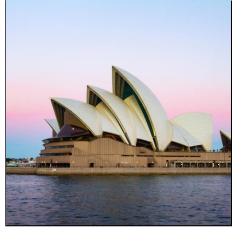
MERU	CLIP
<i>avenida paulista. fisheye 2 kodak elite chrome 100</i>	<i>leica m7 with voigtlander zoom finder and dspth camera strap</i>
<i>rolleiflex kodak portra 160 epson v500 scanner</i>	<i>rollei minidigi</i>
<i>...of my brand new shiny 7.1mp camera.</i>	<i>zeiss ikon icarex, 02/2010</i>
<i>camera de 5mp</i>	<i>faux lomo from www.dumpnr.net - photo fun</i>
<i>i'm a lumix camera fan now.</i>	<i>zeiss-ikon</i>
[ROOT]	[ROOT]



(6)

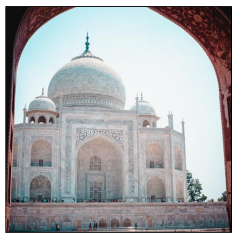
MERU	CLIP
<i>double rainbows in our field were too good to pass up photographing</i>	<i>double rainbows in our field were too good to pass up photographing</i>
<i>whoa... double rainbow</i>	<i>is that... a double rainbow? ;-)</i>
<i>is that... a double rainbow? ;-)</i>	<i>what does it mean!</i>
<i>what does it mean of this picture?</i>	↓
<i>only god could create something so beautiful.</i>	↓
<i>this is a good one to end with. reminds me of the woman in this picture.</i>	↓
<i>look out for that right one.</i>	↓
[ROOT]	[ROOT]

Figure B.2: Image traversals (2/20) using a set of captions from the YFCC dataset.



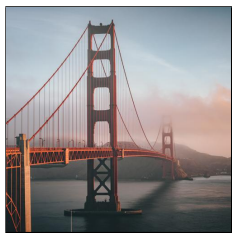
(7)

MERU	CLIP
<i>sydney opera house, october 2012.</i>	<i>gros plan sur l'opera de sydney</i>
<i>sydney opera house see where this picture was taken.</i>	<i>you can just make out the opera house in the far left.</i>
<i>from the new opera house</i>	<i>my sydney</i>
<i>i think this is the last one i have of the opera house.</i>	↓
<i>oh, and some opera house, too.</i>	↓
<i>just next to the famous opera house</i>	↓
<i>from horseshoe bay.</i>	↓
<i>taken from the donau.</i>	↓
[ROOT]	[ROOT]



(8)

MERU	CLIP
<i>captured this during my visit to taj mahal, seems like it still inspires young hearts.....</i>	<i>a weekend adventure to agra to see the taj mahal, see also afternoon and night. luxury.</i>
<i>the royal mausoleum on the grounds of 'iolani palace</i>	<i>this pre-dates the taj mahal</i>
<i><usr> taj</i>	<i>you couldn't photograph inside the tombs, so this is all i can show</i>
<i>rotunda at nmai</i>	<i>the beauty of age, the mark of wisdom</i>
<i>outside of yet another palace</i>	<i>don't remember where.</i>
<i>taken from city palace.</i>	↓
<i>photography ii</i>	↓
<i>it can be seen from most places in the city</i>	↓
<i>kla photography</i>	↓
[ROOT]	[ROOT]



(9)

MERU	CLIP
<i>a high dynamic range shot of the golden gate bridge on a foggy afternoon</i>	<i>golden gate bridge thru photoshop lightroom</i>
<i>working on the perfect golden gate shot.</i>	<i>golden gates</i>
<i>golden gate iii</i>	<i>no...it's not the golden gate ;)</i>
<i>everyone who's been to sf has to take this photo at least once in their lives, right?</i>	<i>by gusf bit.ly/17hga6r</i>
<i>just got back from sf. will post more on my photoblog: ohad.me</i>	<i>the independent sf</i>
<i>i3 sf</i>	<i>thinking about painting this makes my shoulder hurt.</i>
<i>come out and play sf</i>	<i>still searching for the shot around here.</i>
<i>back from golden bay</i>	↓
<i>without fog</i>	↓
[ROOT]	[ROOT]

Figure B.3: Image traversals (3/20) using a set of captions from the YFCC dataset.



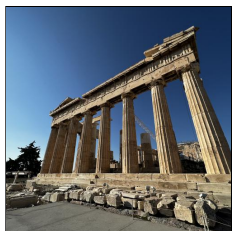
(10)

MERU	CLIP
<i>white cliffs of dover. august, maybe 2004?</i>	<i>coloured sand cliffs of alum bay, isle of wight, 1 may 2012.</i>
<i>calcite cumbria england</i>	<i>the cliffs are made of limestone.</i>
<i>poland rocks.</i>	<i>falkenberg from the south</i>
<i>white point natural area</i>	<i>at juta village</i>
<i>balderstone close</i>	<i>it's pretty rocky there.</i>
<i>nepomuk rocks...</i>	↓
<i>l'eglise de giverny</i>	↓
<i>one point if you can tell me where this was taken.</i>	↓
<i>also some kind of guenon, methinks.</i>	↓
[ROOT]	[ROOT]



(11)

MERU	CLIP
<i>at yellowstone national park there are geyser pools called painted pots because of the colors they exude.</i>	<i>yellowstone - noth entrance</i>
<i>yellowstone - noth entrance</i>	<i>...like i was, how yellowstone got its name</i>
<i>no trip to yellowstone is complete without it</i>	<i>i don't remember where this one was. it was striking.</i>
<i>there are hot springs around here somewhere...</i>	↓
<i>wy'east</i>	↓
<i>many places that were stunning to look at.</i>	↓
<i>there are some special places in the earth. this is one !</i>	↓
<i>with this photo... it's almost like taking a vacation just looking at this.</i>	↓
[ROOT]	[ROOT]



(12)

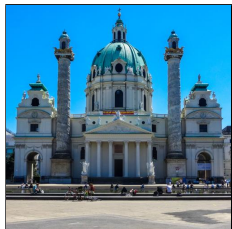
MERU	CLIP
<i>the new parthenon museum, next to the acropolis.</i>	<i>athens archaeological site of the acropolis the parthenon</i>
<i>this is the magnificent temple of zeus, located in the center of athens</i>	<i>temple of jupiter and ruins - selinunte</i>
<i>lil-bit bigger than the athens arch</i>	<i>ruins from roman time</i>
<i>roman building , later used as royal residence .</i>	<i>can't remember where this was</i>
<i>roman fort/settlement.</i>	↓
<i>built in roman times</i>	↓
<i>we will miss our old place.</i>	↓
<i>at quimbleton.</i>	↓
[ROOT]	[ROOT]

Figure B.4: Image traversals (4/20) using a set of captions from the YFCC dataset.



(13)

MERU	CLIP
<i>obligatory big ben shot</i>	<i>big ben i el parlament</i>
<i>it's what they call a 'big clock' could be thought of as 'big ben'</i>	<i>or big ben, to his friends</i>
<i>yeah, i know: big ben is the bell, not the clock tower...</i>	↓
<i>the famous tower</i>	↓
<i>i pretty much only took a photo of this because it was in english</i>	↓
<i>guess what time i took this picture .</i>	↓
[ROOT]	[ROOT]



(14)

MERU	CLIP
<i>karlskirche, just on the outer ring in vienna</i>	<i>kazan, kremin, annunciation cathedral 02/25/2007</i>
<i>vienna, austria - st. charles church</i>	<i>bulgaria, sofia, st. nikolai russian church</i>
<i>wawel cathedral</i>	<i>almost like a cathedral. a cathedral to transit.</i>
<i>near st. george's cathedral</i>	<i>as beautiful as any cathedral</i>
<i>from my old pda.</i>	<i>not far from the cathedral</i>
[ROOT]	[ROOT]



(15)

MERU	CLIP
<i>who needs mt.fuji</i>	<i>fuji provia 100</i>
<i>fuji</i>	<i>fuji-q highlands</i>
<i>mt haba</i>	<i>fuji f30</i>
<i>mount.</i>	<i>fuji-san in the background...</i>
<i>at quimbleton.</i>	<i>fairmount, in</i>
[ROOT]	[ROOT]

Figure B.5: Image traversals (5/20) using a set of captions from the YFCC dataset.



(16)

MERU	CLIP
<i>a single exposure of horseshoe bend at sunrise. certainly one of my favorites from the trip.</i>	<i>yes, there is that backdrop of horseshoe bend :)</i>
<i>horseshoe bend</i>	<i>searching for the one ring</i>
<i>bend over</i>	<i><usr>,usa</i>
<i>looking back at horseshoe canyon</i>	↓
<i>horseshoe bay... this is as close to paradise as you can get!!!!</i>	↓
<i>canyon country, specifically</i>	↓
<i>if you use my photo please post a link and let me know.</i>	↓
[ROOT]	[ROOT]



(17)

MERU	CLIP
<i>the milk way over blieriot ferry provincial park near drumheller, alberta.</i>	<i>the milky way as it appeared above the farmhouse in grey county - 30 sec exposure</i>
<i>the south-western part of the milky way</i>	<i>outside a house from the austmarka region</i>
<i>we were in quite a rural place, although there were still lights on the horizon.</i>	<i>this was just a couple of miles from the farmhouse we stayed in.</i>
<i>keeping the peace while bush was in town</i>	↓
[ROOT]	[ROOT]



(18)

MERU	CLIP
<i>lava as seen through night shot of volcan arenal</i>	<i>lava as seen through night shot of volcan arenal</i>
<i>the majestic villarica volcano</i>	<i>volcan osorno</i>
<i>nice photo of us in front of an active volcano.</i>	<i>volcanic origin</i>
<i>still an active volcano</i>	<i>the volcano!</i>
<i>around the khorgo volcano.</i>	<i>with volcan lanin in the background</i>
<i>mt stromlo</i>	↓
<i>i'm not sure where we were for this shot.</i>	↓
<i>for some reason, i think this photo is great!</i>	↓
[ROOT]	[ROOT]

Figure B.6: Image traversals (6/20) using a set of captions from the YFCC dataset.



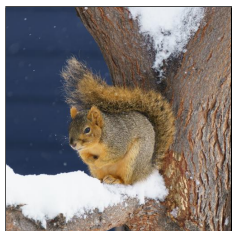
(19)

MERU	CLIP
<i>arcs of the northern lights over the mountains neat troms, norway,</i>	<i>northern lights on the otherside of patreksfjorur</i>
<i>northern lights, norway</i>	<i>aurora boreale a kangerlussuaq</i>
<i>in the night see where this picture was taken.</i>	<i>to a cinema the aurora</i>
<i>from the north. see where this picture was taken.</i>	<i>the village at the end of the world</i>
<i>sometimes something just looks out of place !</i>	<i>over a year's worth of photos here.</i>
[ROOT]	[ROOT]



(20)

MERU	CLIP
<i>cozy cone motel sign with tower of terror in background. california's adventure park at disneyland resort.</i>	<i>adam taylor ollie over sign kodak: iso 200</i>
<i>my favorite tourist attraction in la.</i>	<i>enjoying jason scott's talk.</i>
<i>if this place didn't scream la, i don't know what does.</i>	<i>lost in las vegas- max ruckman</i>
<i>funny, this place was empty.</i>	<i>hollywood rip, ride, rockit</i>
<i>photo : l.g.</i>	↓
[ROOT]	[ROOT]



(21)

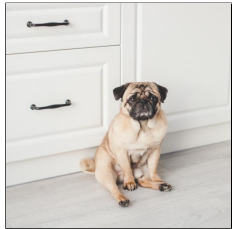
MERU	CLIP
<i>a squirrel enjoying the snow on a not-very cold day.</i>	<i>a squirrel enjoying the snow on a not-very cold day.</i>
<i>winter male still coming to food after the snow.</i>	<i>winter male still coming to food after the snow.</i>
<i>i don't usually see these type of squirrels down here. loved this little guy. :)</i>	↓
<i>the last nut, my dear!</i>	↓
<i>it's kinda fuzzy. but i love this picture for some reason.</i>	↓
<i>i had to take one picture, okay?</i>	↓
[ROOT]	[ROOT]

Figure B.7: Image traversals (7/20) using a set of captions from the YFCC dataset.



(22)

MERU	CLIP
<i>a gull in flight in stratford</i>	<i>a common or arctic tern flying above the scottish peninsula of kintyre.</i>
<i>gull on the wing</i>	<i>more bird shots at dyrholae.</i>
<i>we also saw gull-billed but never close enough to photo.</i>	<i>wouldn't be a trip without at least one picture of a seagull</i>
<i>taken at little gull islands - b, little gull islands</i>	<i>seagull!</i>
<i>taken with the seagull</i>	<i>some bird thing.</i>
<i>i was running after the seagull as i took this photo.</i>	<i>it took patience to get this shot since the stupid bird kept looking away</i>
<i>not a very nice bird, but still interesting to take pictures of..</i>	↓
<i>i took one westward shot, just to see it</i>	↓
<i>i keep taking this photograph</i>	↓
[ROOT]	[ROOT]



(23)

MERU	CLIP
<i>margaret willie sanborn-sebo harvey henry sanborn pug dog framed</i>	<i>sacha my friend and companion patiently waiting for dad to finish taking photos !!!</i>
<i>everyday bear takes up this position as he waits for his mom to make his food.</i>	<i>eli begged mommy to take some photos</i>
<i>patiently waiting for the photographer to get his "shot".</i>	↓
<i>"this picture isn't going to work, and i'm going to show you why..."</i>	↓
<i>thinking to himself, "what's missing?"</i>	↓
<i>this is us... trying to be sultry.</i>	↓
<i>to do nothing or to do something.</i>	↓
[ROOT]	[ROOT]



(24)

MERU	CLIP
<i>zebras, ruaha national park</i>	<i>zebras at kidepo national park, uganda.</i>
<i>zebra fouls with their mohicans, south africa</i>	<i>zebra fouls with their mohicans, south africa</i>
<i>the zebras</i>	<i>zebras at the watering hole</i>
<i>photographer: simone kuipers</i>	<i>three zebras</i>
<i>photographer: mark antos</i>	<i>good things come in threes. apparently, so do zebras.</i>
<i>from photo safari, take 2 .</i>	<i>the group comes around the bend.</i>
<i>at oudja</i>	<i>wild animals</i>
<i>at quimbleton.</i>	<i>more straglers</i>
↓	<i>of the group</i>
↓	<i>just a little to the left of the middle</i>
[ROOT]	[ROOT]

Figure B.8: Image traversals (8/20) using a set of captions from the YFCC dataset.



(25)

MERU	CLIP
<i>monarch, danaus plexippus . shot in waimanalo, hawaii.</i>	<i>a monarch pauses for a drink at a butterfly bush.</i>
<i>taken at the desert botanical garden in phoenix, arizona, during its seasonal monarch butterfly exhibit.</i>	<i>monarch in a standard profile</i>
<i>i'm exercising to capture butterflies.</i>	<i>the monarch</i>
<i>visitor to the butterfly tree,</i>	↓
<i>monarch</i>	↓
<i>butterflies are always free, so enjoy as many of them as you want.</i>	↓
<i>butterfly photography</i>	↓
<i>i tried to get more shots but it flew away.</i>	↓
<i>insect porn</i>	↓
[ROOT]	[ROOT]



(26)

MERU	CLIP
<i>i love the big blooms of the hibiscus with their bold colour.</i>	<i>some hibiscus-like blossoms beside the visitor center at moody gardens in galveston, tx</i>
<i>these looked like hibiscus, but i think they are something else.</i>	<i>after many years nurturing this back to life, the recent heat and rainfall have produced more spectacular blooms.</i>
<i>only a few blooms this time of year...</i>	<i>this beautiful species may be the hibiscus according to my wife but i am not so sure,</i>
<i>much better bloom this time...</i>	<i>these looked like hibiscus, but i think they are something else.</i>
<i>i usually dispise flower photos, but i actually like this one.</i>	<i>looked like they had a lot of nice camera's and video gear. :-)</i>
<i>this is one is good.</i>	<i>flourishing.</i>
[ROOT]	[ROOT]



(27)

MERU	CLIP
<i>crele old english game bantam cockerel</i>	<i>"the beautiful yellow bird - a cautionary tale for new mps" only at acid rabbi.</i>
<i>sabrina is running and twirling to the bachanalia song.</i>	<i>flaunt your assets!</i>
<i>really interesting gadgetar! posted by second life resident ina centaur. visit kaneohe.</i>	<i>most of the girls had gone to college with mary, whose husband was competing.</i>
<i>harvey henry marie elizabeth campbell</i>	↓
<i>sam is showing good form and follow through here.</i>	↓
<i>this is only luky.</i>	↓
[ROOT]	[ROOT]

Figure B.9: Image traversals (9/20) using a set of captions from the YFCC dataset.



(28)

MERU	CLIP
<i>blue-and-yellow macaw at the zooparque itatiba.</i>	<i>i like this portrait of this blue and yellow macaw, because of the black background...</i>
<i>one of my fav birds to shoot at the zoo</i>	<i>from parrot island</i>
<i>i like these birds, especially when you can see the yellow of their eyes!</i>	<i><usr> bird sanctuary.</i>
<i>there are more colorful birds, but there are few birds with as much character.</i>	<i>zero post processing</i>
<i>a beautiful bird, and was quite happy to pose for me.</i>	<i>one of a bunch</i>
<i>never had a bird pose for well before!</i>	↓
<i>hi!some more photos...ana</i>	↓
[ROOT]	[ROOT]



(29)

MERU	CLIP
<i>fly agaric in the forest with a little spider.</i>	<i>freshly popped amanita muscaria in the forest.</i>
<i>all alone on the forest floor</i>	<i>a fly agaric on the rise.</i>
<i>from a recent new forest trip</i>	<i>or "fly agaric". "if your viking gets to choose."</i>
<i>i like this photo, so here it is too.</i>	<i>reminded me a bit of alice in wonderland</i>
↓	<i>reminded me of alice in wonderland</i>
↓	<i>this one goes out to forest love.</i>
[ROOT]	[ROOT]



(30)

MERU	CLIP
<i>taken at mystic aquarium, ct</i>	<i>taken by michael i love watching jellyfish. wish these pics had turned out better.</i>
<i>i don't really like jellyfish, but they are beautiful.</i>	<i>jelly.</i>
<i>something about this reminds me of every photo i've ever taken of jellyfish</i>	↓
<i>it is really very cool to be able to see them...</i>	↓
<i>it did not feel like an aquarium when i took the picture</i>	↓
<i>really not much they let you see here.</i>	↓
<i>that i met.</i>	↓
[ROOT]	[ROOT]

Figure B.10: Image traversals (10/20) using a set of captions from the YFCC dataset.



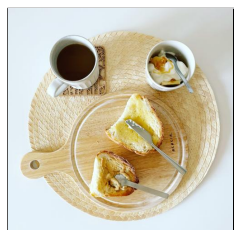
(31)

MERU	CLIP
<i>in new hat from adji</i>	<i>includes my twit hat- wink</i>
<i>oui, girl friend.</i>	<i>in new hat from adji</i>
<i>the hat actually belongs to honey :)</i>	<i>:) he's still in japan right now... i miss him.</i>
<i>fashion photo session</i>	↓
<i>with new hat</i>	↓
<i>a very fit lady.</i>	↓
<i>the boy has style!</i>	↓
[ROOT]	[ROOT]



(32)

MERU	CLIP
<i>humpback coming up for air near juneau, alaska</i>	<i>orca, craig, alaska, tongass nf. usfs francisco sanchez</i>
<i>here, this orca swims about before the show starts.</i>	<i>named for its shape. not for the occasional whales found in its waters.</i>
<i>an orca during the show "believe" in sea world .</i>	<i>a bit shakey!</i>
<i>humpback</i>	<i>takin' off.</i>
<i>a shamu in action</i>	↓
<i>pretend you see a whale and i'll take your photo.</i>	↓
<i>cropitornot shot this one</i>	↓
[ROOT]	[ROOT]



(33)

MERU	CLIP
<i>a pastry & a coffee for breakfast yummy</i>	<i>today's breakfast: toast with honey, egg and black coffee</i>
<i>a right and proper afternoon tea #1</i>	<i>afternoon cream tea</i>
<i>made to perfection. much needed morning tea!</i>	<i>have a nice cup of tea and a sandwich.</i>
<i>some photos from me when i working at home</i>	↓
<i>sobrepeso en la proa</i>	↓
[ROOT]	[ROOT]

Figure B.11: Image traversals (11/20) using a set of captions from the YFCC dataset.



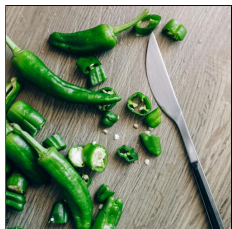
(34)

MERU	CLIP
<i>'tooey's delight' gc16aga</i>	<i>dominic samonte & stephanie estabillo</i>
<i>30 seconds into gina's five-minute toast.</i>	<i>chicken and cheese sandwich.</i>
<i>the plain toast was, um, plain.</i>	↓
<i>with grilled cheese sandwich.</i>	↓
<i>i don't normally like toasted sandwiches, but this one was delicious!</i>	↓
<i>on bread, probably not what you should do with it, but it was a good meal</i>	↓
<i>call center del club cantv</i>	↓
<i>so. good.</i>	↓
[ROOT]	[ROOT]



(35)

MERU	CLIP
<i>at momofuku</i>	<i>where the magic happens at momofuku noodle bar</i>
<i>the ramen got pwned.</i>	<i>@ terakawa ramen 4sq.com/ekseer</i>
<i>siawase ramen</i>	<i>ramen exploration - still looking</i>
<i>this i ate, and it was great!</i>	<i>with michael cotta</i>
<i>i ate some of this.</i>	↓
[ROOT]	[ROOT]



(36)

MERU	CLIP
<i>tiny hot peppers from the freezer</i>	<i>poblanos to be roasted</i>
<i>the ones with jalapenos are the ones that are ruling</i>	<i>they're best as peppers from the garden!</i>
<i>add the chillies and cook for another minute.</i>	<i>do nothing gardening in action!</i>
<i>getting ready for some chili-dippin'</i>	↓
<i>yo tengo el poder!</i>	↓
<i>toying around with a f/1.8</i>	↓
<i>you cannot make this stuff up</i>	↓
[ROOT]	[ROOT]

Figure B.12: Image traversals (12/20) using a set of captions from the YFCC dataset.



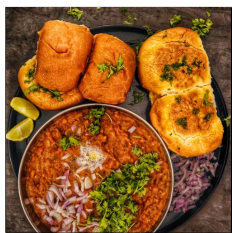
(37)

MERU	CLIP
<i>last nights' salad, cropped</i>	<i>the first of many caprese salads while we traveled.</i>
<i>nice salad, summery, fresh.</i>	<i>for a homegrown salad</i>
<i>contemp salad</i>	<i>another salad :)</i>
<i>i made the salad</i>	↓
<i>it's a verso/kveton thing.</i>	↓
[ROOT]	[ROOT]



(38)

MERU	CLIP
<i>awesome guinness & chocolate cupcakes by kari stewart our wonderful studio manager</i>	<i>i made cupcakes and took a million photos...</i>
<i>peanut butter cupcakes with whipped chocolate ganache.</i>	<i>from the chocolate lady</i>
<i>vegan cupcakes. chocolate, coffee and cinnamon.</i>	↓
<i>coffee and chocolate: a "can't miss" cupcake...</i>	↓
<i>including a flourless chocolate cupcake!</i>	↓
<i>chocolate makes everything better. thanks <usr></i>	↓
<i>chocolate-y goodness!</i>	↓
<i>this week's take, brought over by a friend.</i>	↓
[ROOT]	[ROOT]



(39)

MERU	CLIP
<i>new delhi's best cholle bhature</i>	<i>pav bhaji appetizer, \$5 dinner time only tirupathi bhimas, milpitas</i>
<i>en tlaquepaque, jal.</i>	<i>dal monte la motta .</i>
<i>west indian food. you can't help but smile when you eat it .</i>	<i>the famous curry mile, taken on saturday 5th march</i>
<i><usr> buse dal lof</i>	<i>sloppy everything</i>
<i>paraje guer aike</i>	<i>the midas, great food</i>
<i>manoush</i>	↓
<i>with sambuca</i>	↓
[ROOT]	[ROOT]

Figure B.13: Image traversals (13/20) using a set of captions from the YFCC dataset.



(40)

MERU	CLIP
<i>matcha green tea ice blended</i>	<i>chimichurri sauce recipe</i>
<i>yes i had a pesto drink</i>	<i>trust me, i tried to make this less green.</i>
<i>hot/fermented</i>	<i>make some good ones!</i>
<i>that's right, i've been experimenting. trying to keep things fresh.</i>	↓
<i>ondel-ondel</i>	↓
<i>one love hi pawa</i>	↓
[ROOT]	[ROOT]



(41)

MERU	CLIP
<i>bread with rosemary and garlic infused olive oil at jaleo, a tapas restaurant in my neighborhood.</i>	<i>sliders served at lee roy selmon's restaurant.</i>
<i>with rosemary and parmigiano... our mellow new year's menu</i>	<i>biergarten</i>
<i>bread with olive oil and vinegar</i>	<i>sliders & greens</i>
<i>my welcome brunch to vienna..a cheese party</i>	↓
<i>they serve it with some sort of sauce . this is their version of "bread".</i>	↓
<i>a good shot of how the bread should look.</i>	↓
<i>the bread is real, i think. at least, not glass.</i>	↓
<i>trying out some bread</i>	↓
[ROOT]	[ROOT]



(42)

MERU	CLIP
<i>adam johnston</i>	<i>sparkling apple juice my blog: mikaeladanvers.com</i>
<i>smoked salmon vodka.</i>	<i>nombre sells the accesories</i>
<i>it's so cold in here! look at the frosty vodka</i>	<i>taste like it's fermenting into alcohol.</i>
<i>warm vodka. i'm still cringing.</i>	↓
<i>tasty, tasty cocktails</i>	↓
<i>made with hugin</i>	↓
<i>off for cocktails</i>	↓
[ROOT]	[ROOT]

Figure B.14: Image traversals (14/20) using a set of captions from the YFCC dataset.



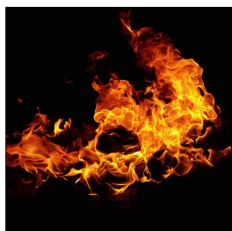
(43)

MERU	CLIP
<i>it feels like winter again #coffeemornings</i>	<i>a latte from blue bottle coffee in oakland, ca.</i>
<i>by phil o'kane aka icedcoffee</i>	<i>sweet cold coffee of destiny especially for <usr> :-) 1 likes on instagram</i>
<i>1 comments on instagram: sorelle.de.latte: you looked great!</i>	↓
<i>heh, maybe my latte art chops aren't so bad after all :-)</i>	↓
<i>cheers <usr> ;)</i>	↓
<i>i likes this one more, i think.</i>	↓
[ROOT]	[ROOT]



(44)

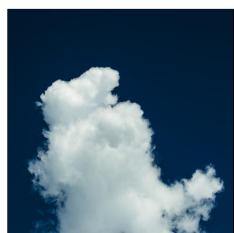
MERU	CLIP
<i>ceu do mapia</i>	<i>caffè ladro at 5 corners in edmonds. americanvirus.com</i>
<i>with a leaf, at the caffè espresso.</i>	<i>chocolate catalan donkey with dinosaur eggs, at eastern time</i>
<i>bourbon & branch</i>	↓
<i>the braan</i>	↓
<i>enjoying the riff raff</i>	↓
<i>b b: the braes</i>	↓
[ROOT]	[ROOT]



(45)

MERU	CLIP
<i>i was well pleased that i managed to capture the flambe moment ;o)</i>	<i>usfws photo/heather webb</i>
<i>everyone was taking pictures of the fire</i>	↓
<i>note to self: need to make more photos of fire again.</i>	↓
<i>this was one of the other interesting things. fire is always fun ;)</i>	↓
<i>i so wanted to photograph this.</i>	↓
[ROOT]	[ROOT]

Figure B.15: Image traversals (15/20) using a set of captions from the YFCC dataset.



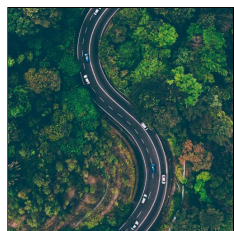
(46)

MERU	CLIP
<i>i caught a cloud! i caught a cloud! *grin*</i>	<i>textures courtesy of shadowhousecreations.blogspot.com/</i>
<i>it's the cloud, baby!</i>	<i>my first cloud photo ever. :)</i>
<i>i liked this cloud.</i>	<i>blowing cloud!</i>
<i>i took this one because of that cloud. seriously.</i>	↓
<i>i like this. i think it liked me.</i>	↓
[ROOT]	[ROOT]



(47)

MERU	CLIP
<i>a drizzly february day in vancouver.</i>	<i>taken with a disposable camera on a rainy seattle day.</i>
<i>weather gloomy all the way to chicago</i>	<i>a cold rainy day in chicago</i>
<i>summer rain in the city.</i>	<i>junechicago</i>
<i>street<usr></i>	↓
<i><usr> street</i>	↓
<i>urban as usual</i>	↓
<i>street pics</i>	↓
[ROOT]	[ROOT]



(48)

MERU	CLIP
<i>this beautiful track goes through deep gorges of nilgiris towards mangalore.</i>	<i>sepang international circuit malaysia</i>
<i>cipularang highway</i>	<i>very well finished road for indonesia!! ... looking down the road</i>
<i><usr> road</i>	<i>nature highway..</i>
<i>kinokuniya just down the road</i>	<i>if you don't know where you are going, any road will take you there.</i>
<i>crookhaven</i>	<i>look, there's a road and everything.</i>
[ROOT]	[ROOT]

Figure B.16: Image traversals (16/20) using a set of captions from the YFCC dataset.



(49)

MERU	CLIP
<i>lebei is driving the quadricycle we rented on the toronto islands.</i>	<i>my cruiser on 80 mile beach. messing with my 15mm zeiss</i>
<i>this moto is still the best anything i've ever owned.</i>	<i>biked to the nearest beach and took some pictures.</i>
<i>you have no idea how much i love this thing.</i>	<i>i was traveling alone, so i instead of me i had to take pictures of my bike</i>
↓	<i>...and she rides like a dream. initial impressions ride report</i>
[ROOT]	[ROOT]



(50)

MERU	CLIP
<i>in a few of the branches of some trees were little tea-light lanterns</i>	<i>in a few of the branches of some trees were little tea-light lanterns</i>
<i>earth hour candle light</i>	<i>they're only lanterns</i>
<i>my first try making this photo with a mini lantern</i>	<i>2nd shoot coming out too</i>
<i>quant little light.</i>	<i>not lit</i>
<i>little lights.</i>	↓
<i>i actually detest gold, but i liked the material contrast here.</i>	↓
<i>i just really liked the light, alright?</i>	↓
<i>a pain to get photos of these.</i>	↓
<i>if you use my photo please post a link and let me know.</i>	↓
[ROOT]	[ROOT]



(51)

MERU	CLIP
<i>simple scandinavian style decor. clifton, bristol</i>	<i>ikea/helsinki design week party</i>
<i>rental apartment in berlin</i>	<i>with jeff from simple plan in denmark</i>
<i>my flat getting more and more comfortable with more furniture coming in - and notice i now have fans!</i>	↓
<i>our first apartment</i>	↓
<i>ready made living space</i>	↓
[ROOT]	[ROOT]

Figure B.17: Image traversals (17/20) using a set of captions from the YFCC dataset.



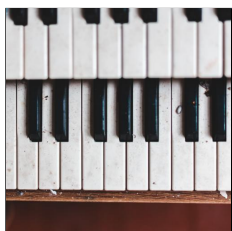
(52)

MERU	CLIP
<i>studio di personaggio. character design. decorated with silver and nickle plated</i>	<i>model: yasemin snoek stylist: melanie vink we bought one of these for a friend's wedding. no, not you julee.</i>
<i>new faucet set.</i>	↓
<i>elegant bath items, though</i>	↓
<i>oh to have that kind of luxury</i>	↓
<i>a little luxury</i>	↓
<i>rich sigfrit kicks things off.</i>	↓
<i>the things i'll do for a shot</i>	↓
<i>can you guess what club this is for?!</i>	↓
[ROOT]	[ROOT]



(53)

MERU	CLIP
<i>pentax *ist ds/ iso 1600 — happy halloween!</i>	<i>jack-o-lantern with other light up decorations. jack-o-lantern.</i>
<i>we wish you a happy all hallows night!</i>	<i>i am so spooooooky</i>
<i>happy hallowe'en, everyone!</i>	↓
<i>happy halloween, yo.</i>	↓
<i>no be long now jack</i>	↓
<i>can you feel the spirit of e-xtrategy?</i>	↓
[ROOT]	[ROOT]



(54)

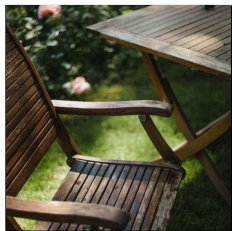
MERU	CLIP
<i>piano keyboard</i>	<i>vintage typewriter photo by rusty blazenhoff</i>
<i>musical ben</i>	<i>the typewriter is the best dead thing i ever found</i>
<i>she was really good, great voice, excellent guitar playing, and really nice to chat to.</i>	<i>musical harmony</i>
<i>the highly-touted prospect, not the guitar player</i>	<i>where some parts of me came of age</i>
<i>l.a.r.g.e</i>	↓
[ROOT]	[ROOT]

Figure B.18: Image traversals (18/20) using a set of captions from the YFCC dataset.



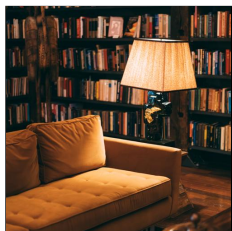
(55)

MERU	CLIP
<i>my christmas shopping is done, and what's more, my presents are all wrapped!</i>	<i>my homemade christmas book - dec. 5th and 6th.</i>
<i>this year's wrapping job.</i>	<i>gift-wrapped for any occasion.</i>
<i>and gift wrapped!</i>	<i>a wrapped xmas present</i>
<i>21 presents for my 21st for msh may</i>	<i>our christmas present put to good use.</i>
<i>christmas is coming. won't someone think of my needs?</i>	↓
<i>the only good thing the guys did was dropped off the gifts.</i>	↓
[ROOT]	[ROOT]



(56)

MERU	CLIP
<i>table in the backyard of the summer house in melby, denmark.</i>	<i>chair detail</i>
<i>kitch at airbnb at corte del correggio - note window behind chairs</i>	↓
<i>photo by laura nawrocik some patio furniture that needs a little cleaning.</i>	↓
<i>from a bench on the north side.</i>	↓
<i>the backyard of the b and b we stayed at</i>	↓
<i>another from this shoot in a more traditional style.</i>	↓
<i>traditional place to take a picture.</i>	↓
<i>a nice place to take pictures!</i>	↓
[ROOT]	[ROOT]



(57)

MERU	CLIP
<i>found at city lights book store, sf www.citylights.com</i>	<i>ikea catalog waiting for pickup, fairborn, ohio.</i>
<i>in powells rare book room</i>	<i>special collections - amsterdam, netherlands</i>
<i>in the rare book reading room</i>	<i>one needs to get there to read the it full.</i>
<i>book heaven.</i>	↓
<i>not a book in sight</i>	↓
<i>even more books</i>	↓
<i>sorry... i really like this space.</i>	↓
<i>with something like this, you would have to get a few</i>	↓
[ROOT]	[ROOT]

Figure B.19: Image traversals (19/20) using a set of captions from the YFCC dataset.



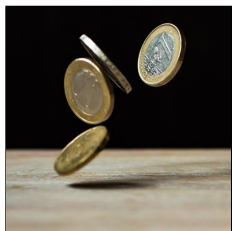
(58)

MERU	CLIP
<i>a pineapple grows in the wild between goa gajah and yeh pulu, bali.</i>	<i>so much organic burden on its way to the sea.</i>
<i>the pineapple, dunmore park. n</i>	<i>everything is so organic on lamu island.</i>
<i>it started with more fruits but i didn't take this picture till late</i>	<i>it's been that long since i took this that the next lot is already growing.</i>
<i>shot <usr> beach</i>	<i>the one that didn't get picked yet</i>
<i>a regular sight from our coast</i>	↓
<i>la nature au carre.</i>	↓
<i>i would be happy if all my photos turned out like this one.</i>	↓
<i>this is one is good.</i>	↓
[ROOT]	[ROOT]



(59)

MERU	CLIP
<i>sulphur crested cockatoos are great characters</i>	<i>soleil when she was a baby with her green feathers</i>
<i>au pied du ciel</i>	<i>lenny white</i>
<i>pale male's mate #5</i>	<i>white tee's - photos for everyone</i>
<i>i think this is a nice photo of sean, though i doubt he will think so.</i>	↓
<i>photo: george struikelblok</i>	↓
<i>this is birdy.</i>	↓
<i>q's and a's</i>	↓
[ROOT]	[ROOT]



(60)

MERU	CLIP
<i>uzi usb drive by dan helmick. brass, wood, riveted.</i>	<i>i swear, this is how the coins landed. so i had to take a photo.</i>
<i>the coin toss</i>	<i>money was a fun picture to take.</i>
<i>inset a coin...it moves...</i>	<i>i really should have pictures of all the money, but after awhile one loses interest.</i>
<i>ver. 2</i>	↓
<i>made in a post secret kinda a way to tell something.</i>	↓
<i>no-one will ever guess</i>	↓
[ROOT]	[ROOT]

Figure B.20: Image traversals (20/20) using a set of captions from the YFCC dataset.

Appendix C

How to Segment and Classify Anything?

C.1 SCAM Implementation Details

In this section, we provide additional details for SCAM, building on our discussion in Section 5.3. Recall that SCAM is composed of four modules: *backbone*, *prompter*, *segmenter*, and *classifier*. We initialize the *backbone* and *classifier* using a pre-trained CLIP model [198], whereas we use SAM [123] (ViT-H [60] checkpoint) as the default *segmenter* in all our experiments.

Pre-trained CLIP models. The versatility of SCAM’s design allows using a variety of CLIP models. Therefore, we experiment with **twelve** publicly available checkpoints:

1. *ResNet-based OpenAI CLIP models* [198]: **Four** models with ResNet image encoders of different sizes [91, 275] – RN50, RN101, RN50-4x, and RN50-16x. These models were trained using 400 million image-text pairs.
2. *ViT-based SigLIP models* [280]: **Five** models trained at 224×224 image resolution and subsequently fine-tuned at higher resolutions: ViT-B-(224px, 256px, 384px), and ViT-L-(256px, 384px) [60]. These models are trained using 10 billion image-text pairs with a *sigmoid loss* instead of the softmax contrastive loss [88].
3. *ConvNeXt-based CLIP models* [110]: **Three** models with ConvNeXt [159] image encoders of different sizes – ConvNeXt-B, ConvNeXt-L, ConvNeXt-XXL. These models are trained using the LAION-2B [211] dataset.

We obtain these checkpoints from the OpenCLIP [110] Model Zoo¹. Next, we elaborate on how we use these models in the *backbone* and *classifier* (Appendices C.1.1 and C.1.2) and further discuss how we train the *prompter* (Appendix C.1.3).

¹Source: https://github.com/mlfoundations/open_clip

C.1.1 Initializing the Backbone with CLIP

Recall that the *backbone* is required to extract dense image embeddings from input images, which are then cropped to create region-level embeddings for the *classifier*. Following the prevailing design, we input 1024×1024 images and obtain 64×64 image embeddings from the backbone.

Primarily, the backbone is used to process the image *once* to amortize the inference cost when classifying image regions which can potentially range between 10–100 or more. This implies a domain shift from the usage of the CLIP image encoder during training, which classifies *whole* images. In our preliminary experiments, we observed that the ConvNeXt-based CLIP models demonstrate superior empirical performance under this domain shift. Therefore, we conduct all experiments in the Section 5.4 using these models. However, we consider all CLIP models in this discussion and report results with ResNet-based and ViT-based CLIP models in Appendix C.2.

ResNet and ConvNeXt have a *hierarchical* design – an input image is initially downsampled to embeddings of $\frac{1}{4}$ scale, then passed through four *convolutional stages*. Each stage progressively reduces the image embedding scale by half until it reaches the final $\frac{1}{32}$ scale compared to the input image. Hence, to obtain image embeddings with the desired $\frac{1}{16}$ scale, we initialize the *backbone* with all stages but the last one. We resize the input images to a maximum size of 1024 pixels (maintaining the aspect ratio), then convert them into a square using zero-padding.

ViT, on the other hand, has an *isotropic* design – they downsample the input images to $\frac{1}{16}$ scale in the first layer, and this scale remains constant throughout all layers. As such, we use the full image encoder in the *backbone* for SigLIP models, excluding the final *attention pooling* layer. To use the ViT with higher-resolution input images, we upsample the positional embeddings via *bilinear* interpolation. Unlike convolutional CLIP models, we *do not* zero-pad the input images – CLIP image encoder does not observe zero-padded patches during pre-training. Zero-padded patches interfere with self-attention in ViT and yield empirically lower results when adapted without fine-tuning.

Note on other ViT-based CLIP models: Besides the CLIP models that we considered above, OpenCLIP contains several publicly available checkpoints using the ViT image encoder. These encoders contain a *learnable* [class] token that aggregates spatial (*patch*) embeddings at each layer. After the final layer, this token is matched with text embeddings

during image-text pre-training. Such a design does not guarantee that the region-level embeddings align well with text embeddings. Zero-shot adaptation of such models is beyond the scope of this study. We encourage further works to develop ViT-based CLIP models without the [class] token. This consideration would enhance the models’ adaptability to region-level tasks and reduce computational overhead on modern hardware like TPU pods [279].

C.1.2 Initializing the Classifier with CLIP

The *classifier* includes the remaining parameters of the CLIP image encoder that are excluded from the *backbone*. For convolutional models, this consists of the final *convolutional stage*, whereas for ViT-based models, the last *attention pooling* layer. These modules process cropped image embeddings, which are obtained using the masks from the *segmenter*. Notably, we crop region embeddings of size $\frac{S}{16} \times \frac{S}{16}$, where S represents the image resolution used during CLIP pre-training. However, we use larger region crops (24×24) for ConvNeXt-L and ConvNeXt-XL models, since they give slightly better empirical results. Utilizing masks from the *segmenter*, we perform masked pooling to obtain foreground-specific embeddings.

The *classifier*’s second part is a weight matrix representing text embeddings. We utilize the six most effective ImageNet templates as prescribed by Radford et al. [198], and we fill them with object class names to extract text embeddings from the matching CLIP-text encoder. In summary, we measure the pairwise similarity between pairs of region-text embeddings, scale it with the learned CLIP temperature (also adding *bias* term for SigLIP), and apply softmax to produce a probability distribution across object classes.

C.1.3 Training the Prompter

The prompter is a lightweight convolutional module that predicts a real-valued score $\in [0, 1]$ for each location in the grid of image embeddings from the *backbone*. The purpose of this module is to propose a subset of points from a dense 128×128 point grid to prompt the *segmenter* (SAM).

Trivial method: binary classification. A trivial method to train the prompter is to predict a binary score – a feature location is assigned 1 (*foreground*) if the underlying image pixel belongs to any ground-truth segmentation mask, and 0 (*background*) otherwise. This strategy resembles the *objectness* objective of the region proposal networks [204].

While effective, this strategy leads to wasted computation and slower runtime due to the inability to choose a subset of points for a particular object. SAM can predict the desired object mask for most points inside the corresponding object of interest. Hence, prompting multiple points for an object may yield *near-duplicate* masks from SAM and hence hurts runtime. Therefore, it becomes imperative to leverage this effectiveness of SAM to improve runtime – not only should the *prompter* generate point prompts with *high recall* (covering *all* objects of interest), but also with *high precision* (very few points for each object).

Training with mask-based centerness: To obtain point prompts with higher precision, we train the prompter to predict real-valued scores $\in [0, 1]$ denoting the *centerness* of a feature location. With a well-trained prompter, one can sample more point prompts closer to object centers than boundaries. The centerness objective is used in existing detectors, *e.g.*, CenterNet [289] and FCOS [238], however, they define this measure using bounding boxes instead of masks.

We use the mean-squared error (MSE) to train the *prompter*. During training, every feature location is assigned a ground-truth regression target computed using unlabeled masks. For every instance mask of a given image, we compute the Euclidean distance transform (EDT) [206] at 128×128 resolution. Then we scale the target values to $[0, 1]$ and subsequently combine the distance transforms of all instances by computing element-wise maximum. Note that we ignore the *padding* locations from loss computation for non-square images. Finally, we avoid points close to object boundaries by assigning targets $\in [0, 0.25]$ to background (0), then re-scaling targets back to $\in [0, 1]$. At test-time, we find *peaks* in the heatmap of logits using a 3×3 max-pooling kernel.

C.2 Zero-shot Transfer with SCAM

Table C.1 shows zero-shot transfer results for SCAM using different types of CLIP models. When transferred for object detection, CLIP image encoders face a domain mismatch as they operate on high-resolution images (1024 pixels) as compared to much lower resolution during pre-training (typically 224–384 pixels). In this scenario, we observe that convolutional CLIP models fare better than ViT-based counterparts. Moreover, ConvNeXt-based models show stronger empirical performance than ResNet-based models, likely due to higher parameter count and training data size. Among SigLIP models, we observe that ViT models fine-tuned with higher resolution inputs perform better, as the domain gap characterized by image resolution is reduced by high-resolution fine-tuning. Based on

CLIP Image Encoder	Params	COCO	
		AP	AP ₅₀
ResNet-based OpenAI CLIP Models			
RN50	38M	11.4	17.1
RN101	56M	11.0	16.3
RN50-4x	87M	12.4	18.5
RN50-16x	167M	13.6	19.8
ViT-based SigLIP Models			
ViT-B-224px	92M	10.0	15.1
ViT-B-256px	92M	11.7	17.5
ViT-B-384px	92M	15.9	24.1
ViT-L-256px	316M	13.8	20.5
ViT-L-384px	316M	16.9	25.1
ConvNeXt-based OpenCLIP Models			
ConvNeXt-B	88M	21.9	32.7
ConvNeXt-L	200M	24.8	37.9
ConvNeXt-XXL	846M	25.8	39.1

Table C.1: **Zero-shot transfer to COCO, using different backbones with SCAM.** All models use our test-time improvements (Mask-based NMS, Sub-mask Suppression, and masked pooling). ConvNeXt-based models perform the best, which we use throughout our remaining experiments.

these results, we use the ConvNeXt-based models by default in our experiments.

Figures C.1 and C.2 show more zero-shot transfer results of SCAM with the ConvNeXt-XXL backbone. We obtain reasonable mask predictions without any additional fine-tuning. While the detector performance is far from current state-of-the-art models (that train on these datasets), our results suggest that SCAM can aid in object discovery and can bootstrap learning via iterative self-training.

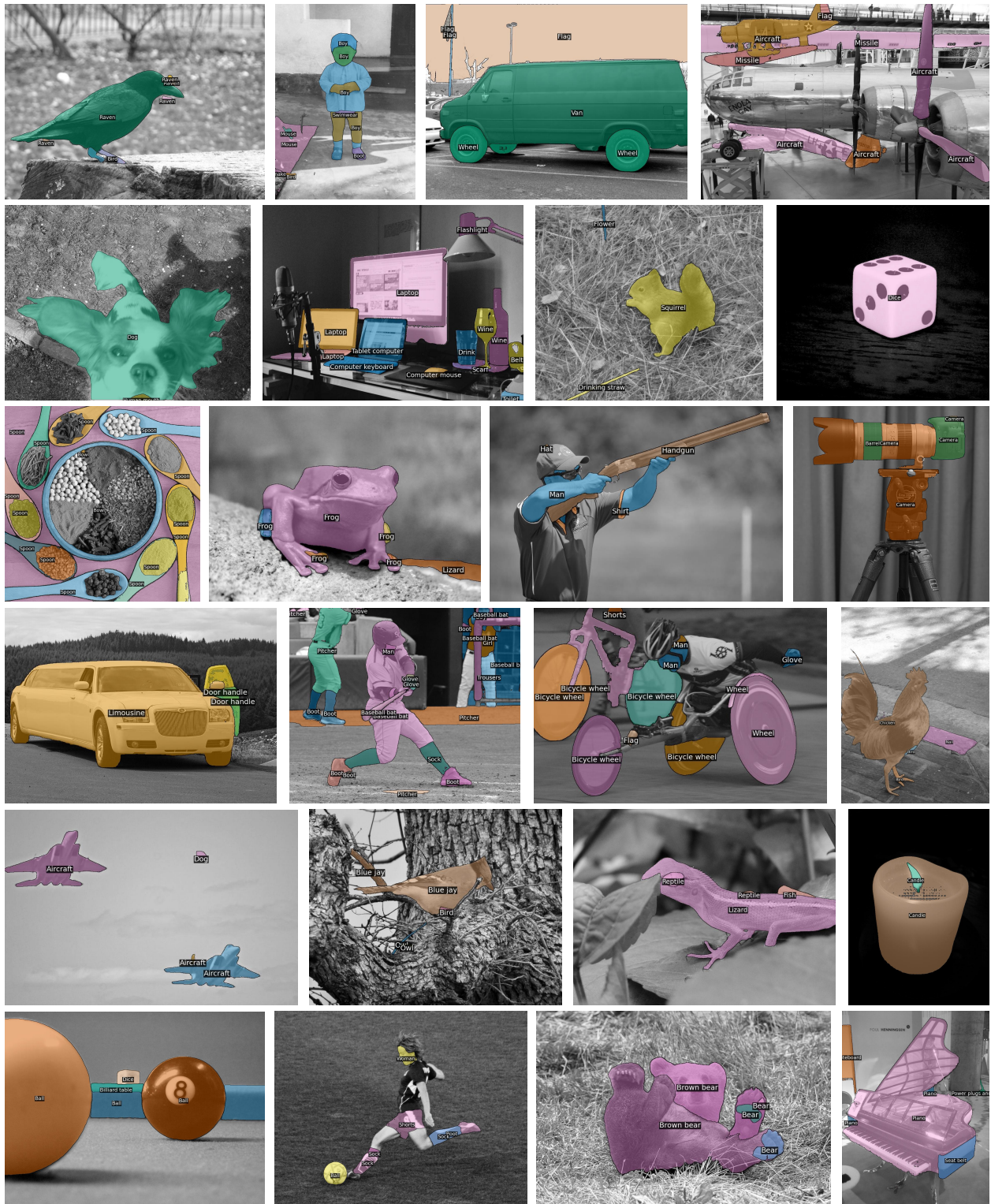


Figure C.1: **Zero-shot transfer with SCAM: Qualitative results.** High-scoring masks predicted by SCAM with CLIP ConvNeXt-XXL for *random* images from OpenImages [16]. SCAM can segment novel objects without *any* downstream fine-tuning.

Bibliography

- [1] Harsh Agrawal, Karan Desai, Yufei Wang, Xinlei Chen, Rishabh Jain, Mark Johnson, Dhruv Batra, Devi Parikh, Stefan Lee, and Peter Anderson. nocaps: novel object captioning at scale. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2019. [13](#), [14](#), [25](#)
- [2] Rami Al-Rfou, Marc Pickett, Javier Snaider, Yun hsuan Sung, and Brian Strope. Conversational Contextual Cues: The Case of Personalization and History for Response Ranking. *arXiv preprint arXiv:1606.00372*, 2016. [28](#)
- [3] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. SPICE: Semantic propositional image caption evaluation. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2016. [20](#), [38](#)
- [4] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. [24](#)
- [5] Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton Van Den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. [2](#)
- [6] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. VQA: Visual question answering. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2015. [2](#), [6](#), [24](#), [25](#), [38](#)
- [7] 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. [55](#), [65](#)
- [8] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. [10](#), [35](#)
- [9] Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning Representations by Maximizing Mutual Information Across Views. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. [24](#)

- [10] Alexei Baevski and Michael Auli. Adaptive Input Representations for Neural Language Modeling. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019. [35](#)
- [11] 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. [65](#)
- [12] Ankan Bansal, Karan Sikka, Gaurav Sharma, Rama Chellappa, and Ajay Divakaran. Zero-shot object detection. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2018. [69](#)
- [13] Michael Barthel, Galen Stocking, Jesse Holcomb, and Amy Mitchell. Nearly Eight in Ten Reddit Users get News on the Site. *Pew Research Center*, 2016. [31](#)
- [14] Jason Baumgartner, Savvas Zannettou, Brian Keegan, Megan Squire, and Jeremy Blackburn. The Pushshift Reddit Dataset. *arXiv preprint arXiv:2001.08435*, 2020. [28](#), [105](#)
- [15] Emily M Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? In *Proceedings of the ACM Conference on Fairness, Accountability, and Transparency*, 2021. [29](#), [30](#), [31](#)
- [16] Rodrigo Benenson, Stefan Popov, and Vittorio Ferrari. Large-scale interactive object segmentation with human annotators. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. [xi](#), [xii](#), [xiv](#), [67](#), [76](#), [77](#), [134](#)
- [17] Thomas Berg, Jiongxin Liu, Seung Woo Lee, Michelle L. Alexander, David W. Jacobs, and Peter N. Belhumeur. Birdsnap: Large-scale Fine-grained Visual Categorization of Birds. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. [36](#)
- [18] Abeba Birhane and Vinay Uday Prabhu. Large Image Datasets: A Pyrrhic Win for Computer Vision? In *Proceedings of IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2021. [4](#), [29](#), [30](#), [31](#)
- [19] Abeba Birhane, Vinay Uday Prabhu, and Emmanuel Kahembwe. Multimodal datasets: misogyny, pornography, and malignant stereotypes. *arXiv preprint arXiv:2110.01963*, 2021. [83](#)
- [20] Vighnesh Birodkar, Zhichao Lu, Siyang Li, Vivek Rathod, and Jonathan Huang. The surprising impact of mask-head architecture on novel class segmentation. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2021. [70](#)
- [21] Lukas Bossard, M. Guillaumin, and L. Gool. Food-101 - Mining Discriminative Components with Random Forests. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2014. [36](#), [102](#), [105](#)

- [22] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020. [9](#), [20](#), [26](#), [29](#)
- [23] Mert Bulent Sariyildiz, Julien Perez, and Diane Larlus. Learning Visual Representations with Caption Annotations. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2020. [15](#), [16](#), [25](#), [38](#), [64](#), [82](#)
- [24] Joy Buolamwini and Timnit Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Proceedings of the ACM Conference on Fairness, Accountability, and Transparency*, 2018. [4](#), [31](#)
- [25] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2020. [5](#), [69](#), [70](#), [75](#)
- [26] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep Clustering for Unsupervised Learning of Visual Features. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2018. [7](#), [24](#)
- [27] Mathilde Caron, Piotr Bojanowski, Julien Mairal, and Armand Joulin. Unsupervised pre-training of image features on non-curated data. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2019. [7](#)
- [28] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised Learning of Visual Features by Contrasting Cluster Assignments. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. [12](#), [15](#), [16](#), [24](#), [35](#)
- [29] Mathilde Caron, Hugo Touvron, Ishan Misra, Herv'e J'egou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging Properties in Self-Supervised Vision Transformers. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2021. [37](#)
- [30] 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. [4](#), [26](#), [29](#), [32](#), [38](#), [50](#), [87](#)
- [31] Tianqi Chen, Bing Xu, Chiyuan Zhang, and Carlos Guestrin. Training deep nets with sublinear memory cost. *arXiv preprint arXiv:1604.06174*, 2016. [18](#), [104](#)
- [32] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A Simple Framework for Contrastive Learning of Visual Representations. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2020. [3](#), [7](#), [24](#), [35](#)

- [33] Xi Chen, Zhiyan Zhao, Yilei Zhang, Manni Duan, Donglian Qi, and Hengshuang Zhao. Focalclick: Towards practical interactive image segmentation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 72
- [34] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015. 8, 11, 12, 25, 34, 38, 51
- [35] 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. 50, 51
- [36] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Learning universal image-text representations. *arXiv preprint arXiv:1909.11740*, 2019. 16, 24, 25, 38
- [37] Bowen Cheng, Alexander G. Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 69, 73, 75
- [38] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 5, 69
- [39] Gong Cheng, Junwei Han, and Xiaoqiang Lu. Remote Sensing Image Scene Classification: Benchmark and State of the Art. *Proceedings of the IEEE*, 2017. 102
- [40] Francois Chollet. Xception: Deep Learning with Depthwise Separable Convolutions. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2, 39
- [41] Mircea Cimpoi, Subhansu 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. 102
- [42] Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. ELEC-TRA: Pre-training Text Encoders as Discriminators Rather Than Generators. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020. URL <https://openreview.net/pdf?id=r1xMH1BtvB>. 9
- [43] 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. 102
- [44] Navneet Dalal and Bill Triggs. Histograms of Oriented Gradients for Human Detection. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005. 2

- [45] 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. [65](#)
- [46] Terrance de Vries, Ishan Misra, Changhan Wang, and Laurens van der Maaten. Does object recognition work for everyone? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019. [4](#), [29](#)
- [47] 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. [1](#), [6](#), [7](#), [25](#), [52](#), [69](#), [102](#)
- [48] Jia Deng, Olga Russakovsky, Jonathan Krause, Michael S Bernstein, Alex Berg, and Li Fei-Fei. Scalable multi-label annotation. In *SIGCHI*, 2014. [8](#)
- [49] Jiankang Deng, Jia Guo, Zhou Yuxiang, Jinke Yu, Irene Kotsia, and Stefanos Zafeiriou. RetinaFace: Single-stage Dense Face Localisation in the Wild. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. [30](#)
- [50] 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. [4](#), [34](#), [35](#), [37](#), [38](#), [64](#)
- [51] 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. [4](#), [50](#), [87](#), [105](#)
- [52] Karan Desai, Maximilian Nickel, Tanmay Rajpurohit, Justin Johnson, and Ramakrishna Vedantam. Hyperbolic Image-Text Representations. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2023. [5](#)
- [53] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics – Human Language Technologies (NAACL HLT)*, 2019. [9](#), [10](#), [11](#), [16](#), [24](#), [82](#)
- [54] Edsger W Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1959. [57](#)
- [55] Jesse Dodge, Andreea Gane, Xiang Zhang, Antoine Bordes, Sumit Chopra, Alexander Miller, Arthur Szlam, and Jason Weston. Evaluating Prerequisite Qualities for Learning End-to-End Dialog Systems. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2016. [28](#)
- [56] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised Visual Representation Learning by Context Prediction. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2015. [2](#), [7](#), [23](#), [105](#)

- [57] Jeff Donahue and Karen Simonyan. Large Scale Adversarial Representation Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. [24](#)
- [58] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2014. [6](#)
- [59] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. [6](#), [9](#)
- [60] 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. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021. [44](#), [50](#), [64](#), [68](#), [69](#), [71](#), [83](#), [129](#)
- [61] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2015. [2](#)
- [62] Albert Einstein. Zur Elektrodynamik bewegter Körper. *Annalen der physik*, 1905. [45](#)
- [63] 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. [45](#)
- [64] Mohamed El Banani, Karan Desai, and Justin Johnson. Learning Visual Representations via Language-Guided Sampling. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. [83](#), [106](#)
- [65] 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. [52](#), [64](#), [105](#)
- [66] Aleksandr Ermolov, Leyla Mirvakhabova, Valentin Khrukov, Nicu Sebe, and Ivan Osleedets. Hyperbolic vision transformers: Combining improvements in metric learning. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. [65](#)
- [67] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John M. Winn, and Andrew Zisserman. The Pascal Visual Object Classes (VOC) Challenge. *IJCV*, 2009. [12](#), [16](#), [25](#), [38](#)

- [68] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. LIBLINEAR: A library for large linear classification. *Journal of machine learning research*, 2008. [12](#)
- [69] 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. [102](#)
- [70] 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. [64](#)
- [71] Andreas Fürst, Elisabeth Rumetshofer, Johannes Lehner, Viet T Tran, Fei Tang, Hubert Ramsauer, David Kreil, Michael Kopp, Günter Klambauer, Angela Bitto, et al. Cloob: Modern hopfield networks with infoloob outperform clip. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. [106](#)
- [72] Samir Yitzhak Gadre, Gabriel Ilharco, Alex Fang, Jonathan Hayase, Georgios Smyrnis, Thao Nguyen, Ryan Marten, Mitchell Wortsman, Dhruva Ghosh, Jieyu Zhang, Eyal Orgad, Rahim Entezari, Giannis Daras, Sarah M Pratt, Vivek Ramanujan, Yonatan Bitton, Kalyani Marathe, Stephen Mussmann, Richard Vencu, Mehdi Cherti, Ranjay Krishna, Pang Wei Koh, Olga Saukh, Alexander Ratner, Shuran Song, Hananeh Hajishirzi, Ali Farhadi, Romain Beaumont, Sewoong Oh, Alex Dimakis, Jenia Jitsev, Yair Carmon, Vaishaal Shankar, and Ludwig Schmidt. Datacomp: In search of the next generation of multimodal datasets. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. [83](#)
- [73] Oran Gafni, Adam Polyak, Oron Ashual, Shelly Sheynin, Devi Parikh, and Yaniv Taigman. Make-a-scene: Scene-based text-to-image generation with human priors. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2022. [83](#)
- [74] Octavian-Eugen Ganea, Gary Bécigneul, and Thomas Hofmann. Hyperbolic Entailment Cones for Learning Hierarchical Embeddings. *arXiv preprint arXiv:1804.01882*, 2018. [48](#), [65](#), [100](#)
- [75] 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. [65](#)
- [76] Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé III, and Kate Crawford. Datasheets for datasets. *arXiv preprint arXiv:1803.09010*, 2018. [29](#), [31](#), [87](#)
- [77] Sam Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A Smith. Re-alsoxicityprompts: Evaluating neural toxic degeneration in language models. *arXiv preprint arXiv:2009.11462*, 2020. [30](#)

- [78] Golnaz Ghiasi, Yin Cui, Aravind Srinivas, Rui Qian, Tsung-Yi Lin, Ekin D Cubuk, Quoc V Le, and Barret Zoph. Simple copy-paste is a strong data augmentation method for instance segmentation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 69, 71
- [79] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018. 2, 7, 24
- [80] Ross Girshick. Fast R-CNN. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2015. 69, 74, 76
- [81] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 2, 6, 69, 76
- [82] Ross Girshick, Ilija Radosavovic, Georgia Gkioxari, Piotr Dollár, and Kaiming He. Detectron. <https://github.com/facebookresearch/detectron>, 2018. 17
- [83] Priya Goyal, Dhruv Mahajan, Abhinav Gupta, and Ishan Misra. Scaling and Benchmarking Self-Supervised Visual Representation Learning. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 7, 11, 12
- [84] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 24
- [85] 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. 52, 69
- [86] Agrim Gupta, Piotr Dollar, and Ross Girshick. LVIS: A dataset for large vocabulary instance segmentation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. xi, xii, xiv, 7, 16, 17, 25, 37, 38, 67, 68, 76, 77, 135
- [87] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2010. 3, 24
- [88] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality Reduction by Learning an Invariant Mapping. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006. 24, 129
- [89] Alon Halevy, Peter Norvig, and Fernando Pereira. The Unreasonable Effectiveness of Data. *IEEE Intelligent Systems*, 2009. URL http://www.computer.org/portal/cms_docs_intelligent/intelligent/homepage/2009/x2exp.pdf. 66

- [90] Cong Han, Yujie Zhong, Dengjie Li, Kai Han, and Lin Ma. Zero-shot semantic segmentation with decoupled one-pass network. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2023. 69
- [91] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2, 4, 6, 9, 19, 25, 34, 38, 69, 129
- [92] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2017. 2, 5, 6, 17, 19, 37, 68, 69, 70, 72, 73, 74, 75, 80
- [93] Kaiming He, Ross Girshick, and Piotr Dollár. Rethinking ImageNet pre-training. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2019. 13, 69
- [94] 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. 7, 12, 13, 15, 16, 24, 35
- [95] 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. 3, 16, 65
- [96] 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. 51, 69
- [97] 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. 102
- [98] Olivier J Hénaff, Ali Razavi, Carl Doersch, SM Eslami, and Aaron van den Oord. Data-efficient Image Recognition with Contrastive Predictive Coding. *arXiv preprint arXiv:1905.09272*, 2019. 7, 24
- [99] Matthew Henderson, Paweł Budzianowski, Iñigo Casanueva, Sam Coope, Daniela Gerz, Girish Kumar, Nikola Mrkšić, Georgios Spithourakis, Pei-Hao Su, Ivan Vulic, and Tsung-Hsien Wen. A Repository of Conversational Datasets. In *Proceedings of the Workshop on NLP for Conversational AI*, 2019. 28
- [100] Lisa Anne Hendricks, Kaylee Burns, Kate Saenko, Trevor Darrell, and Anna Rohrbach. Women also snowboard: Overcoming bias in captioning models. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2018. 31
- [101] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016. 10

- [102] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the Knowledge in a Neural Network. *NeurIPS Deep Learning and Representation Learning Workshop*, 2015. URL <http://arxiv.org/abs/1503.02531>. 2, 39
- [103] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning Deep Representations by Mutual Information Estimation and Maximization. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019. 24
- [104] Micah Hodosh, Peter Young, and Julia Hockenmaier. Framing image description as a ranking task: Data, models and evaluation metrics. *JAIR*, 2013. 8
- [105] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The Curious Case of Neural Text Degeneration. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020. 37
- [106] Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. *spaCy: Industrial-strength Natural Language Processing in Python*, 2020. URL <https://doi.org/10.5281/zenodo.1212303>. 34, 57
- [107] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2
- [108] Zhicheng Huang, Zhaoyang Zeng, Bei Liu, Dongmei Fu, and Jianlong Fu. Pixel-BERT: Aligning Image Pixels with Text by Deep Multi-Modal Transformers. *arXiv preprint arXiv:2004.00849*, 2020. 25, 38
- [109] Drew A Hudson and Christopher D Manning. GQA: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 24, 25, 38
- [110] 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>. 50, 129
- [111] Hakan Inan, Khashayar Khosravi, and Richard Socher. Tying word vectors and word classifiers: A loss framework for language modeling. *arXiv preprint arXiv:1611.01462*, 2016. 10
- [112] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2015. 11
- [113] Grant Jenkins. Python Word Segmentation. URL <https://github.com/grantjenkins/python-wordsegment>. 35

- [114] 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. [26](#), [32](#), [38](#), [43](#), [64](#), [69](#), [82](#)
- [115] Eun Seo Jo and Timnit Gebru. Lessons from archives: Strategies for collecting sociocultural data in machine learning. In *Proceedings of the ACM Conference on Fairness, Accountability, and Transparency*, 2020. [29](#)
- [116] 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. [102](#)
- [117] 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. [6](#), [9](#), [11](#)
- [118] 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. [51](#), [64](#)
- [119] Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. Referitgame: Referring to objects in photographs of natural scenes. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014. [24](#), [38](#)
- [120] Valentin Khruikov, 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. [55](#), [65](#)
- [121] Wonjae Kim, Bokyung Son, and Ildoo Kim. ViLT: Vision-and-Language Transformer Without Convolution or Region Supervision. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2021. [38](#)
- [122] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017. [35](#)
- [123] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment Anything. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2023. [5](#), [68](#), [69](#), [70](#), [72](#), [76](#), [129](#)
- [124] Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Big transfer (bit): General visual representation learning. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2020. [2](#), [21](#)

- [125] Simon Kornblith, Jonathon Shlens, and Quoc V Le. Do better imagenet models transfer better? In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 106
- [126] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3D Object Representations for Fine-Grained Categorization. In *2013 IEEE International Conference on Computer Vision Workshops*, 2013. 36, 102
- [127] Jonathan Krause, Justin Johnson, Ranjay Krishna, and Li Fei-Fei. A Hierarchical Approach for Generating Descriptive Image Paragraphs. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2
- [128] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, Michael S Bernstein, and Li Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *IJCV*, 2017. 24
- [129] Ranjay A Krishna, Kenji Hata, Stephanie Chen, Joshua Kravitz, David A Shamma, Li Fei-Fei, and Michael S Bernstein. Embracing error to enable rapid crowdsourcing. In *CHI*, 2016. 8
- [130] Alex Krizhevsky. Learning Multiple Layers of Features from Tiny Images. 2009. URL <https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf>. 102
- [131] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2012. 1, 6, 38, 64, 69
- [132] Taku Kudo and John Richardson. SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing. In *EMNLP: System Demonstrations*, 2018. 11, 35
- [133] Weicheng Kuo, Yin Cui, Xiuye Gu, AJ Piergiovanni, and Anelia Angelova. Open-Vocabulary Object Detection upon Frozen Vision and Language Models. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2023. 69
- [134] Aditya Kusupati, Gantavya Bhatt, Aniket Rege, Matthew Wallingford, Aditya Sinha, Vivek Ramanujan, William Howard-Snyder, Kaifeng Chen, Sham M. Kakade, Praateek Jain, and Ali Farhadi. Matryoshka Representation Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 44, 54
- [135] Gant Laborde. Deep NN for NSFW Detection. https://github.com/GantMan/nsfw_model. 30
- [136] Hugo Laurençon, Lucile Saulnier, Léo Tronchon, Stas Bekman, Amanpreet Singh, Anton Lozhkov, Thomas Wang, Siddharth Karamcheti, Alexander M Rush, Douwe Kiela, et al. Obelisc: An open web-scale filtered dataset of interleaved image-text documents. *arXiv preprint arXiv:2306.16527*, 2023. 83

- [137] 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. 47
- [138] *Our List of Dirty, Naughty, Obscene, and Otherwise Bad Words*. LDNOOBW Github organization. <https://github.com/LDNOOBW/List-of-Dirty-Naughty-Obscene-and-Otherwise-Bad-Words>. 30
- [139] 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)*, 2019. 47, 48, 49, 65
- [140] Yann LeCun, Corinna Cortes, and CJ Burges. MNIST handwritten digit database. *ATT Labs [Online]*. Available: <http://yann.lecun.com/exdb/mnist>, 2010. 102
- [141] 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>. 43, 44, 100
- [142] 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. 21, 36, 38, 64
- [143] Gen Li, Nan Duan, Yuejian Fang, Daxin Jiang, and Ming Zhou. Unicoder-VL: A universal encoder for vision and language by cross-modal pre-training. *AAAI Conference on Artificial Intelligence*, 2020. 16, 24, 25, 38
- [144] Jingyao Li, Pengguang Chen, Shengju Qian, and Jiaya Jia. Tagclip: Improving discrimination ability of open-vocabulary semantic segmentation. *arXiv preprint arXiv:2304.07547*, 2023. 69
- [145] Junnan Li, Pan Zhou, Caiming Xiong, Richard Socher, and Steven C.H. Hoi. Prototypical Contrastive Learning of Unsupervised Representations. *arXiv preprint arXiv:2005.04966*, 2020. 15, 16, 24
- [146] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. BLIP: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2022. 83
- [147] Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. VisualBERT: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*, 2019. 16, 24, 25, 38
- [148] Wen Li, Limin Wang, Wei Li, Eirikur Agustsson, and Luc Van Gool. WebVision Database: Visual Learning and Understanding from Web Data. *arXiv preprint arXiv:1708.02862*, 2017. 39

- [149] Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, et al. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2020. [16](#), [24](#), [25](#), [38](#)
- [150] 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 Pre-training Paradigm. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2022. [50](#), [52](#), [106](#)
- [151] Yanghao Li, Hanzi Mao, Ross Girshick, and Kaiming He. Exploring plain vision transformer backbones for object detection. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2022. [5](#), [70](#), [71](#)
- [152] Feng Liang, Bichen Wu, Xiaoliang Dai, Kunpeng Li, Yanan Zhao, Hang Zhang, Peizhao Zhang, Peter Vajda, and Diana Marculescu. Open-vocabulary semantic segmentation with mask-adapted clip. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. [69](#), [76](#)
- [153] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2014. [xiv](#), [7](#), [13](#), [16](#), [25](#), [37](#), [38](#), [68](#), [76](#), [77](#), [78](#), [79](#)
- [154] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. [17](#), [69](#)
- [155] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2017. [69](#)
- [156] Dong C Liu and Jorge Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical programming*, 1989. [106](#)
- [157] Qin Liu, Zhenlin Xu, Gedas Bertasius, and Marc Niethammer. Simpleclick: Interactive image segmentation with simple vision transformers. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2023. [72](#)
- [158] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv preprint arXiv:1907.11692*, 2019. [9](#), [20](#), [57](#)
- [159] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. [69](#), [78](#), [129](#)

- [160] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 2, 6
- [161] Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 11, 13, 35, 51
- [162] Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019. 35, 51
- [163] David G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision (IJCV)*, 2004. 2
- [164] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. ViLBERT: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 16, 24, 25, 38
- [165] Jiasen Lu, Christopher Clark, Sangho Lee, Zichen Zhang, Savya Khosla, Ryan Marten, Derek Hoiem, and Aniruddha Kembhavi. Unified-io 2: Scaling autoregressive multimodal models with vision, language, audio, and action. *arXiv preprint arXiv:2312.17172*, 2023. 83
- [166] Dhruv Kumar Mahajan, Ross B. Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten. Exploring the Limits of Weakly Supervised Pretraining. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2018. 2, 13, 21, 39
- [167] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew B. Blaschko, and Andrea Vedaldi. Fine-Grained Visual Classification of Aircraft. *arXiv preprint arXiv:1306.5151*, 2013. 102
- [168] Junhua Mao, Jiajing Xu, Kevin Jing, and Alan L Yuille. Training and evaluating multimodal word embeddings with large-scale web annotated images. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016. 8, 20, 24
- [169] David Marr. *Vision: A computational investigation into the human representation and processing of visual information*. MIT Press, 1982. 1, 67
- [170] James Martens and Ilya Sutskever. Training deep and recurrent networks with hessian-free optimization. In *Neural networks: Tricks of the trade*, 2012. 18
- [171] Pierre-Emmanuel Mazaré, Samuel Humeau, Martin Raison, and Antoine Bordes. Training Millions of Personalized Dialogue Agents. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018. 28
- [172] 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. 11, 51, 104

- [173] George A Miller. WordNet: A Lexical Database for English. In *Speech and Natural Language: Proceedings of a Workshop Held at Harriman, New York*, 1992. 52
- [174] Hermann Minkowski. Raum und Zeit. *Physikalische Zeitschrift*, 1908. 45
- [175] Ishan Misra and Laurens van der Maaten. Self-Supervised Learning of Pretext-Invariant Representations. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 7, 12, 24
- [176] 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. xv, 44, 50, 51, 52, 102, 104, 105
- [177] Kevin P. Murphy. *Machine learning : a probabilistic perspective*. MIT Press, 2013. 43, 53
- [178] 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. 65
- [179] Maximilian Nickel and Douwe Kiela. Poincaré Embeddings for Learning Hierarchical Representations. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. 43, 65
- [180] 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. 47, 55
- [181] Maria-Elena Nilsback and Andrew Zisserman. Automated Flower Classification over a Large Number of Classes. In *Indian Conference on Computer Vision, Graphics and Image Processing*, 2008. 36, 102
- [182] Mehdi Noroozi and Paolo Favaro. Unsupervised Learning of Visual Representations by Solving Jigsaw Puzzles. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2016. 2, 24, 105
- [183] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation Learning with Contrastive Predictive Coding. *arXiv preprint arXiv:1807.03748*, 2018. 7, 24
- [184] OpenAI. GPT-4V(ision) System Card. https://cdn.openai.com/papers/GPTV_System_Card.pdf, 2023. 83
- [185] Vicente Ordonez, Girish Kulkarni, and Tamara L. Berg. Im2Text: Describing Images Using 1 Million Captioned Photographs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2011. 8, 20, 24, 25, 38, 87
- [186] 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. 36, 102, 105

- [187] 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. [11](#), [50](#), [58](#)
- [188] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature Learning by Inpainting. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. [2](#), [7](#), [24](#)
- [189] Amandalynne Paullada, Inioluwa Deborah Raji, Emily M Bender, Emily Denton, and Alex Hanna. Data and its (dis) contents: A survey of dataset development and use in machine learning research. *arXiv preprint arXiv:2012.05345*, 2020. [29](#), [30](#)
- [190] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12, 2011. [12](#), [36](#), [106](#)
- [191] Chao Peng, Tete Xiao, Zeming Li, Yuning Jiang, Xiangyu Zhang, Kai Jia, Gang Yu, and Jian Sun. MegDet: A large mini-batch object detector. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. [17](#)
- [192] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics – Human Language Technologies (NAACL HLT)*, 2018. [9](#), [10](#)
- [193] Boris T Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 1964. [11](#)
- [194] Ofir Press and Lior Wolf. Using the output embedding to improve language models. In *EACL*, 2017. [10](#)
- [195] Di Qi, Lin Su, Jia Song, Edward Cui, Taroon Bharti, and Arun Sacheti. ImageBERT: Cross-modal Pre-training with Large-scale Weak-supervised Image-Text Data. *arXiv preprint arXiv:2001.07966*, 2020. [38](#)
- [196] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. *OpenAI blog*, 2018. [9](#), [10](#), [11](#), [26](#)
- [197] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 2019. [9](#), [11](#), [20](#), [26](#), [29](#), [30](#), [35](#)

- [198] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning Transferable Visual Models From Natural Language Supervision. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2021. [xv](#), [5](#), [26](#), [27](#), [29](#), [32](#), [35](#), [36](#), [38](#), [43](#), [46](#), [48](#), [50](#), [52](#), [53](#), [64](#), [65](#), [68](#), [69](#), [70](#), [82](#), [102](#), [103](#), [104](#), [105](#), [106](#), [129](#), [131](#)
- [199] Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollár. Designing network design spaces. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. [2](#)
- [200] 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>. [44](#)
- [201] *Reddit: the front page of the internet*. Reddit, . <https://reddit.com>. [26](#)
- [202] *Reddit API*. Reddit, . <https://www.reddit.com/dev/api>. [28](#)
- [203] *Python Reddit API Wrapper v7.1.0*. Reddit, . <https://github.com/praw-dev/praw>. [28](#)
- [204] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2015. [17](#), [69](#), [74](#), [76](#), [131](#)
- [205] Lawrence G. Roberts. *Machine Perception of Three-Dimensional Solids*. PhD thesis, Massachusetts Institute of Technology, 1963. [1](#)
- [206] Azriel Rosenfeld and John L Pfaltz. Distance functions on digital pictures. *Pattern recognition*, 1(1):33–61, 1968. [73](#), [132](#)
- [207] 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. [54](#)
- [208] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 2015. [1](#), [6](#), [7](#), [12](#), [25](#), [38](#)
- [209] Axel Sauer, Tero Karras, Samuli Laine, Andreas Geiger, and Timo Aila. Stylegan-t: Unlocking the power of gans for fast large-scale text-to-image synthesis. *arXiv preprint arXiv:2301.09515*, 2023. [83](#)
- [210] 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. [5](#), [50](#), [83](#)

- [211] 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. In *NeurIPS Datasets and Benchmarks*, 2022. [5](#), [43](#), [50](#), [54](#), [83](#), [129](#)
- [212] 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. [43](#)
- [213] 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. [11](#), [35](#), [50](#)
- [214] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. CNN features off-the-shelf: an astounding baseline for recognition. In *CVPR Workshops*, 2014. [6](#)
- [215] 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. [4](#), [8](#), [20](#), [24](#), [25](#), [26](#), [29](#), [38](#), [50](#), [87](#)
- [216] Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. Megatron-LM: Training multi-billion parameter language models using model parallelism. *arXiv preprint arXiv:1909.08053*, 2019. [9](#), [20](#)
- [217] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. [2](#)
- [218] 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. [54](#), [102](#)
- [219] Konstantin Sofiiuk, Ilya A Petrov, and Anton Konushin. Reviving iterative training with mask guidance for interactive segmentation. In *2022 IEEE International Conference on Image Processing (ICIP)*, 2022. [72](#)
- [220] Kihyuk Sohn. Improved Deep Metric Learning with Multi-class N-pair Loss Objective. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016. [48](#), [64](#)
- [221] Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep Metric Learning via Lifted Structured Feature Embedding. *arXiv preprint arXiv:1511.06452*, 2015. [64](#)

- [222] Robyn Speer. ftfy, 2019. URL <https://doi.org/10.5281/zenodo.2591652>. 29, 106
- [223] Krishna Srinivasan, Karthik Raman, Jiecao Chen, Michael Bendersky, and Marc Najork. WIT: Wikipedia-based Image Text Dataset for Multimodal Multilingual Machine Learning. *arXiv preprint arXiv:2103.01913*, 2021. 32, 38
- [224] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research (JMLR)*, 2014. 10
- [225] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. VL-BERT: Pre-training of generic visual-linguistic representations. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020. 16, 24, 25, 38
- [226] Alane Suhr, Stephanie Zhou, Ally Zhang, Iris Zhang, Huajun Bai, and Yoav Artzi. A corpus for reasoning about natural language grounded in photographs. In *Proceedings of the Annual Meeting on Association for Computational Linguistics (ACL)*, 2019. 24, 25, 38
- [227] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting Unreasonable Effectiveness of Data in Deep Learning Era. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2017. 2, 21, 66
- [228] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2013. 11
- [229] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 2
- [230] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2, 30, 35
- [231] Hao Tan and Mohit Bansal. LXMERT: Learning cross-modality encoder representations from transformers. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2019. 16, 24, 25, 38
- [232] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2019. 2
- [233] H Tankovska. Distribution of Reddit.com traffic 2020, by country, April 2021. URL <https://www.statista.com/statistics/325144/reddit-global-active-user-distribution/>. 31

- [234] Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023. [83](#)
- [235] David Thiel and Jeffrey Hancock. Identifying and eliminating csam in generative ml training data and models. *Stanford Digital Repository*, 2023. [83](#)
- [236] 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. [21](#), [38](#), [39](#), [50](#), [58](#), [104](#), [106](#)
- [237] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2020. [7](#), [24](#)
- [238] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. FCOS: A simple and strong anchor-free object detector. In *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2020. [69](#), [73](#), [132](#)
- [239] Alexandru Tifrea, Gary Bécigneul, and Octavian-Eugen Ganea. Poincaré GloVe: Hyperbolic Word Embeddings. *arXiv preprint arXiv:1810.06546*, 2018. [65](#)
- [240] 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. [50](#)
- [241] Trieu H Trinh, Minh-Thang Luong, and Quoc V Le. Selfie: Self-supervised Pretraining for Image Embedding. *arXiv preprint arXiv:1906.02940*, 2019. [24](#)
- [242] Michael Tschannen, Manoj Kumar, Andreas Steiner, Xiaohua Zhai, Neil Houlsby, and Lucas Beyer. Image captioners are scalable vision learners too. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. [83](#)
- [243] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The inaturalist species classification and detection dataset. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. [16](#), [25](#), [38](#)
- [244] 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. [4](#), [7](#), [10](#), [24](#), [34](#), [35](#), [43](#), [50](#)
- [245] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. CIDEr: Consensus-based image description evaluation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. [20](#), [38](#)

- [246] Bastiaan S Veeling, Jasper Linmans, Jim Winkens, Taco Cohen, and Max Welling. CNNs for Digital Pathology. *arXiv preprint arXiv:1806.03962*, 2018. 54, 102
- [247] 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. 43, 64
- [248] 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. 65
- [249] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 2, 4, 6, 9, 11
- [250] Michael Völske, Martin Potthast, Shahbaz Syed, and Benno Stein. TL;DR: Mining Reddit to Learn Automatic Summarization. In *Proceedings of the Workshop on New Frontiers in Summarization, ACL*, 2017. 26, 28
- [251] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge J. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. 2011. 102
- [252] Qiang Wang, Bei Li, Tong Xiao, Jingbo Zhu, Changliang Li, Derek F. Wong, and Lidia S. Chao. Learning Deep Transformer Models for Machine Translation. In *Proceedings of the Annual Meeting on Association for Computational Linguistics (ACL)*, 2019. 35
- [253] Xinlong Wang, Tao Kong, Chunhua Shen, Yuning Jiang, and Lei Li. SOLO: Segmenting objects by locations. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2020. 69
- [254] Yuqing Wang, Zhaoliang Xu, Hao Shen, Baoshan Cheng, and Lirong Yang. Centermask: single shot instance segmentation with point representation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 69
- [255] Ross Wightman. PyTorch Image Models. <https://github.com/rwightman/pytorch-image-models>, 2019. 50
- [256] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. <https://github.com/facebookresearch/detectron2>, 2019. 13, 16
- [257] Zhirong Wu, Yuanjun Xiong, Stella Yu, and Dahua Lin. Unsupervised Feature Learning via Non-parametric Instance-level Discrimination. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 3, 24, 37, 65
- [258] 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. 36

- [259] 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. [102](#)
- [260] Qizhe Xie, Eduard Hovy, Minh-Thang Luong, and Quoc V Le. Self-training with Noisy Student improves ImageNet classification. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. [21](#)
- [261] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. [2](#), [19](#), [69](#)
- [262] Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tie-Yan Liu. On Layer Normalization in the Transformer Architecture. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2020. [35](#)
- [263] Jiarui Xu, Shalini De Mello, Sifei Liu, Wonmin Byeon, Thomas Breuel, Jan Kautz, and Xiaolong Wang. Groupvit: Semantic segmentation emerges from text supervision. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. [83](#)
- [264] Mengde Xu, Zheng Zhang, Fangyun Wei, Han Hu, and Xiang Bai. Side adapter network for open-vocabulary semantic segmentation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. [69](#)
- [265] I Zeki Yalniz, Hervé Jégou, Kan Chen, Manohar Paluri, and Dhruv Mahajan. Billion-scale Semi-supervised Learning for Image Classification. *arXiv preprint arXiv:1905.00546*, 2019. [2](#), [21](#)
- [266] Kaiyu Yang, Jacqueline Yau, Li Fei-Fei, Jia Deng, and Olga Russakovsky. A Study of Face Obfuscation in ImageNet. *arXiv preprint arXiv:2103.06191*, 2021. [29](#), [30](#)
- [267] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. XLNet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. [9](#)
- [268] 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. [50](#), [52](#)
- [269] Mang Ye, Xu Zhang, Pong C Yuen, and Shih-Fu Chang. Unsupervised Embedding Learning via Invariant and Spreading Instance Feature. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. [24](#)

- [270] 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. [51](#), [64](#)
- [271] Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *TACL*, 2014. [8](#), [24](#)
- [272] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models. *arXiv preprint arXiv:2205.01917*, 2022. [83](#)
- [273] Qihang Yu, Ju He, Xueqing Deng, Xiaohui Shen, and Liang-Chieh Chen. Convolutions die hard: Open-vocabulary segmentation with single frozen convolutional clip. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. [69](#)
- [274] Seonghoon Yu, Paul Hongsuck Seo, and Jeany Son. Zero-shot referring image segmentation with global-local context features. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. [69](#)
- [275] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016. [19](#), [129](#)
- [276] 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. [52](#), [69](#)
- [277] Rowan Zellers, Yonatan Bisk, Ali Farhadi, and Yejin Choi. From recognition to cognition: Visual commonsense reasoning. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. [24](#), [25](#), [38](#)
- [278] Xiaohua Zhai, Joan Puigcerver, Alexander Kolesnikov, Pierre Ruysen, 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. [102](#)
- [279] Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. [131](#)
- [280] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2023. [129](#)

- [281] Michael Zhang, James Lucas, Jimmy Ba, and Geoffrey E Hinton. Lookahead Optimizer: k steps forward, 1 step back. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. [11](#)
- [282] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful Image Colorization. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2016. [2](#), [7](#), [23](#), [105](#)
- [283] Richard Zhang, Phillip Isola, and Alexei A Efros. Split-Brain Autoencoders: Unsupervised Learning by Cross-Channel Prediction. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. [2](#), [7](#), [23](#)
- [284] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. [2](#)
- [285] Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. Men also like shopping: Reducing gender bias amplification using corpus-level constraints. *arXiv preprint arXiv:1707.09457*, 2017. [31](#)
- [286] Yiwu Zhong, Jianwei Yang, Pengchuan Zhang, Chunyuan Li, Noel Codella, Lianian Harold Li, Luowei Zhou, Xiyang Dai, Lu Yuan, Yin Li, and Jianfeng Gao. RegionClip: Region-based Language-image Pretraining. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. [69](#)
- [287] Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Learning Deep Features for Scene Recognition using Places Database. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2014. [25](#)
- [288] Luowei Zhou, Hamid Palangi, Lei Zhang, Houdong Hu, Jason J Corso, and Jianfeng Gao. Unified vision-language pre-training for image captioning and VQA. *AAAI Conference on Artificial Intelligence*, 2020. [16](#), [24](#), [25](#), [38](#)
- [289] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019. [69](#), [132](#)
- [290] Wanrong Zhu, Jack Hessel, Anas Awadalla, Samir Yitzhak Gadre, Jesse Dodge, Alex Fang, Youngjae Yu, Ludwig Schmidt, William Yang Wang, and Yejin Choi. Multimodal c4: An open, billion-scale corpus of images interleaved with text. *arXiv preprint arXiv:2304.06939*, 2023. [83](#)
- [291] Yuke Zhu, Oliver Groth, Michael Bernstein, and Li Fei-Fei. Visual7w: Grounded question answering in images. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. [2](#), [6](#), [24](#), [25](#), [38](#)
- [292] Chengxu Zhuang, Alex Lin Zhai, and Daniel Yamins. Local Aggregation for Unsupervised Learning of Visual Embeddings. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2019. [24](#)