Using Knowledge Graphs to harvest datasets for efficient CLIP model training

Simon Ging Sebastian Walter Jelena Bratulić Johannes Dienert Hannah Bast Thomas Brox

University of Freiburg, Germany
gings@cs.uni-freiburg.de
https://github.com/lmb-freiburg/entitynet

Abstract

Training high-quality CLIP models typically requires enormous datasets, which limits the development of domain-specific models – especially in areas that even the largest CLIP models do not cover well – and drives up training costs. This poses challenges for scientific research that needs fine-grained control over the training procedure of CLIP models. In this work, we show that by employing smart web search strategies enhanced with knowledge graphs, a robust CLIP model can be trained from scratch with considerably less data. Specifically, we demonstrate that an expert foundation model for living organisms can be built using just 10M images. Moreover, we introduce EntityNet, a dataset comprising 33M images paired with 46M text descriptions, which enables the training of a generic CLIP model in significantly reduced time.

1. Introduction

Contrastive Language-Image Pretraining (CLIP) [29] has become a cornerstone for training Vision-Language Models (VLMs). CLIP models learn high-quality visual embeddings and establish a link to the semantic level of brief text descriptions by training on pairs of images and their corresponding text descriptions collected from the web. The features and the link between images and text have been used directly for, e.g., zero-shot classification or text-to-image retrieval, and enable dialogues with visual input, such as in the LLaVA family of models [20–22]. The link can also be exploited in the opposite direction to enable text-conditional image generation, e.g., Stable Diffusion [28].

Training state-of-the-art CLIP models is computationally expensive. The original CLIP model has seen 12.8B image-text pairs, and later works have scaled this further [8, 10]. This need for scale has limited most of the research to finetuning, which comes with reduced architectural flexibility and control over the data selection. It is particularly problematic for analytic research that demands full control over training to find causes of emergent behavior.

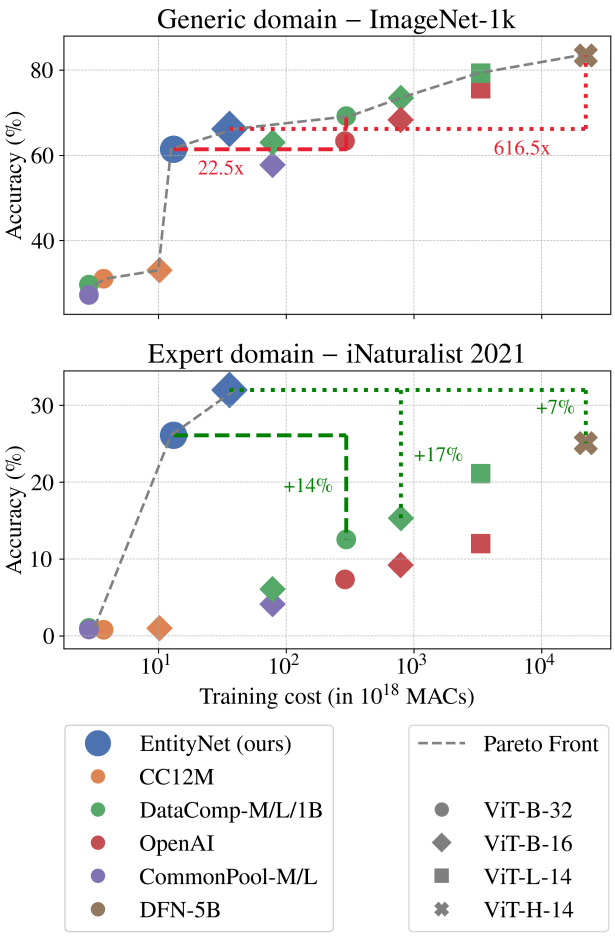


Figure 1. We demonstrate how to harvest datasets for training CLIP models with an improved quality-cost trade-off, either for a generic domain (top) or an expert domain (bottom).

The effort to collect vast datasets is also a key bottleneck for building foundation models for expert domains. Although CLIP models are supposed to be generic and cover

most of the world, they are not good enough for use in specific expert domains such as medicine or biology. Building foundation models for expert domains requires an efficient data collection process, taking into account the availability of fewer data samples in these domains.

Our goal is to tackle these challenges from the dataset side while keeping the CLIP algorithm fixed. This strategy is backed by recent literature. For example, Li et al. [18] explored CLIP "along three dimensions: data, architecture, and training strategies" and they stress the "significance of high-quality training data". For Large Language Models (LLMs), data curation was shown to reduce training time and model size, achieved through heavily filtered publicly available web data and synthetic data [2]. With the dataset creation process, we aim (1) for improved performance in the expert domain of living organisms, in order to demonstrate the creation of expert foundation models; and (2) we aim for a good trade-off between training efficiency and model performance on the broad domain of the visual world, in order to enable compute-efficient from-scratch analysis of fully functional CLIP models.

We built a dataset we named *EntityNet*, where we leveraged knowledge graphs and targeted web image search. Specifically, from the knowledge graphs Wikidata and WordNet, we collected 135k entities (e.g. *eagle*) as well as their aliases and descriptions. We extracted entity attributes from Wikidata related to color, partonomy, behavior, and other aspects, which we then used to guide an LLM in generating entity-attribute queries for image search.. For example, from the entity *plastic* and the attribute *small* we generated the search term *small plastic item*.

The resulting EntityNet consists of 33M images paired with 45M alt texts and 613k text labels from the knowledge graphs. The dataset is partitioned into a subset of 10M images of living organisms, capturing high-quality visual and semantic information about the taxonomy of animals, plants, and funghi, as well as a subset of 23M images covering a wide range of categories, such as tools, geographical features, materials, and buildings. Notably, from this process we obtain not only alt texts, but also a link back to the knowledge graph information that was used to create the search query for a given image. We show that this information can be used during training to achieve better performance than by training on alt texts alone. The method of creating our dataset is largely generic and can be applied to other knowledge graphs.

Training on this dataset, we obtain a foundation model that is both specialized on the target expert domain and is also able to understand the overall visual world. In our domain-specific evaluations on iNaturalist and RareSpecies, the model demonstrates robust generalization and clearly surpasses CLIP models trained on much more data (Figure 1). On ImageNet, we demonstrate our dataset to be highly compute efficient and to achieve a performance comparable to models trained 20x longer (Figure 1).

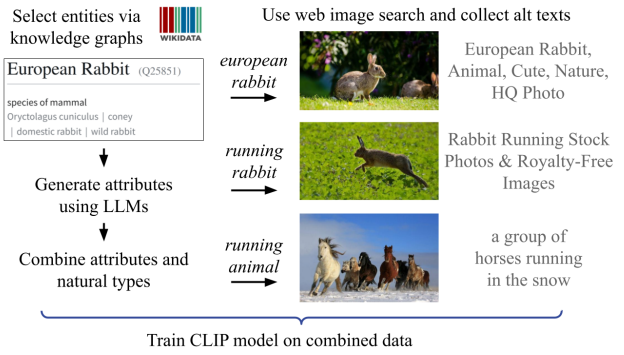


Figure 2. We create a dataset for vision-language pretraining: First, we extract entities from knowledge graphs, then generate attributes and natural types for them. We search for different combinations of entities, attributes, and types in image search engines, and collect alt texts for each image. Finally, we train our model on the combined data.

- We propose a method to automatically create a vision-language dataset based on a given knowledge graph and an image search engine.
- We apply this method to create the *EntityNet* dataset, consisting of 33M images paired with 45M alt texts and supplementary text information from the knowledge graphs.
- We train an expert CLIP model for living organisms on a single 8xL40S machine from scratch in 55 hours. This *EntityNet-CLIP* is highly specialized in the target expert domain of living organisms, *and* comparably strong on ImageNet.
- We evaluate our model and a suite of other CLIP models for object classification, image retrieval, and domain shift robustness. In the expert domain of animals and plants, our model achieves higher performance than models with orders of magnitude more parameters or training data. It is also much stronger than CLIP models that specialize only for this domain. In the generic domain, our model performs remarkably well given the low amount of compute required to train it.

2. Related work

Datasets. Many recent works investigated ways to build large-scale datasets for multimodal training. Radford et al. [29] train the original CLIP model on a closed set of 400M images, with the model weights being released, but not the data. They build their dataset by collecting image-text pairs, where the text includes frequent terms derived from Wikipedia or WordNet nouns, and approximately balance the classes in the result. As a first approach to create a public dataset of this size, Schuhmann et al. [33]

built a dataset with 400 million image-text pairs by filtering HTML data from Common Crawl (LAION-400M) [30]. Their main method of filtering is to remove all image-text pairs that have less than 0.3 similarity estimated by the CLIP model. In a follow-up work, they [34] scaled their approach up one order of magnitude with the multilingual LAION-5B dataset. Xu et al. [48] intended to replicate the original CLIP's data curation approach. They collected image-text pairs from CommonCrawl and filtered them using Wikipedia and WordNet, then balanced the results. Gadre et al. [10] proposed DataComp, a filtering challenge with a candidate pool of up to 13B image-text pairs from CommonCrawl, where the goal is to filter this candidate pool and to run a fixed training pipeline on the resulting data. They proposed a baseline DataComp-1B dataset with 1.4B pairs filtered with a combination of CLIP score and clustering CLIP embeddings to find images close to ImageNet [7] training examples. Fang et al. [8] trained a Data Filtering Network on an internal dataset of 357M human-verified image-text pairs and finetuned it on a set of public, human-annotated datasets. They filtered 42B candidates into the DFN-5B dataset and trained the current top model of the OpenCLIP leaderboard [14].

These large datasets have mostly replaced smaller datasets like ConceptualCaptions12M (CC12M) [4], which relies on unimodal heuristics and Google Cloud Vision APIs to predict image-text similarity. Another popular small dataset is Yahoo Flickr Creative Commons 15M (YFCC15M), a subset of 15M image-text pairs obtained from the YFCC100M dataset which is based on Flickr [38]. Although many works are mostly concerned with scaling up multimodal datasets and models as much as possible, we aim to improve research on high-quality CLIP models in scenarios where data and compute efficiency is important, for example, for setting up a CLIP model for an expert domain or for scientific analysis of CLIP training.

Stevens et al. [35] aimed for a general vision model named BioCLIP for organismal biology and curated the TreeOfLife-10M dataset based on the Encyclopedia of Life [1], iNaturalist 2021 [41] and BIOSCAN-1M [11]. BioCLIP is trained on a mix of english and latin entity names. For evaluation, they curated the RareSpecies benchmark, which tests generalization to 400 species unseen during training. While the authors of BioCLIP used biological domain knowledge to build their dataset, we rather rely on knowledge graphs and propose a dataset collection method for arbitrary domains.

Training algorithms. Various works are concerned with improving CLIP from the algorithmic side. Li et al. [16] simply train on low resolution first, then finetune on higher resolution later. In another work [17], the authors additionally mask a substantial portion of the image to further reduce the amount of input during training. Zhai et al. [51] propose using a sigmoid loss which reduces the computational load especially in big distributed settings. They follow up [39] by extending the training objective using multiple previously developed techniques, including captioning-based pretraining [45], self-distillation [24] and online data curation [40] into one unified recipe. Vasu et al. [42] enhance their training data with synthetic captions created by an image captioning model and use an ensemble of CLIP teachers to train their model. This way, they can increase the learning efficiency by transferring knowledge from bigger models to their smaller models. Chen et al. [5] evaluate the choice of vision encoder and design a hybrid architecture that improves over vanilla vision transformers when trained under the CLIP framework. Such algorithmic improvements are orthogonal to our research, since they would potentially also improve training on our dataset. In this work, we focus on data improvements and fix the algorithm and architecture choices, since this also allows to easily and fairly compare to a big set of already trained Vision Transformer (ViT) based CLIP models.

Li et al. [18] scale down CLIP and analyse the influence of different data, architecture, and training strategies. They find that especially large models need larger datasets, and data quality plays an important role. They create higher quality datasets by applying CLIP filtering to the 3.4B WebLI dataset [6], while we aim to use a different dataset collection process.

3. Dataset creation

In this section, we outline our dataset creation process, consisting of four main steps: entity extraction, attribute generation, query building, and image search. This process is generally applicable to all visual domains covered by the underlying knowledge graph. In this work, we construct a dataset covering the majority of common visual entities in our world. Within this broad domain, we particularly focus on animals and plants, referred to as the *living* subset in this section. See Figure 2 for an illustration of the dataset creation process.

3.1. Entity extraction

A high-quality list of visual entities forms the basis for our dataset. We build our list of entities from the Wikidata knowledge graph [43], taking advantage of the hierarchical structure provided by the *subclass of* relation within Wikidata. For example, the entity *dog* is a subclass of the *pet* entity, which in turn is a subclass of *domesticated animal* entity. Through this hierarchy, we can easily collect all entities under a particular governing or super-entity. First, we manually build a list of 21 super-entities that cover most physical and visual entities in Wikidata. For the *living* subset, the super-entities are just *animal* and *plant*. Examples of non-living super-entities are *food*, *building*, or *physical tool*,

Table 1. Examples of entities and additional information as extracted from the Wikidata knowledge graph. Name, description, and aliases are used as text labels during training. The number of sitelinks are considered a proxy for an entity's popularity. The name and aliases are used during search.

Entity	Description	Sitelinks	Aliases
tiger	species of big cat	216	tigress, tigers, Panthera tigris
chest	box-shaped type of furniture	51	coffer, kist
muscle car	type of high-performance car	30	high performance car

with all super-entities listed in the supplementary material. Next, we extract all entities from Wikidata that are linked to at least one of the super-entities through the *subclass of* relation. For animals and plants, Wikidata also models their biological taxon hierarchy via the *parent taxon* relation. Because the taxon hierarchy substantially increases the coverage of our *living* subset, we use it together with the regular *subclass* hierarchy to extract *living* entities. Note that we do not want to include named entities (e.g., specific persons). Since Wikidata models these via the *instance of* relation, our extraction process excludes them by using only the *subclass of* and *parent taxon* relations. For every entity, we also download its name, description, aliases, and its number of Wikimedia sitelinks [1] as additional information. See Section 3.1 for examples of entities. Finally, we apply two filtering steps: First, we remove all entities with a sitelink count below a predefined threshold, eliminating very rare or low-quality entities which are unlikely to produce strong search results. We find that this filtered list of entities still contains a significant amount of non-visual entities. To address this, we use a LLM to classify each entity as visual or non-visual, discarding the non-visual ones.

For our expert domain, the *living* subset, we also add all nouns from WordNet [9] that are a subclass of the *living thing* node, excluding humans, named entities and entities that cannot be seen with the bare eye, e.g., microorganisms.

Finally, we employ heuristic methods to detect and remove potentially offensive entities via a profanity filter.

3.2. Attribute generation

Besides searching for the entities directly, we also aim to search for variations of them in different contexts, by combining them with attributes. We manually define 6 visual attribute categories we want to search for: *Color*, *Pattern and texture*, *Parts*, *Shape and size*, *Environment*, and *Other*. We extract potential attributes for each entity from the Wikidata knowledge graph and prompt an LLM [2] with this entity and attribute information to generate a list of visual attributes. For each attribute we also generate a search query combining the attribute itself with the corresponding entity. See Section 3.2 for examples of generated attributes and search queries. We generate between 1 and 10 attributes per category and generate them for the most popular entities only, as image search engines fail to respect attributes in search queries for rare entities, where they often even struggle to return good results for the entity alone.

Table 2. Examples of attributes and corresponding search queries for different entities as generated by the LLM. Search queries are used for image search and during training.

Entity	Category	Attribute	Search query
rock	Pattern and texture	porous	porous rock
wolf	Environment	snow	wolf in the snow
residence	Parts	arches	arches in residence architecture
garlic	Shape and size	big	big garlic bulb
farm	Other	tourist	tourists visiting a farm
boot	Color	multicolor	multicolored boots

3.3. Query building

For the entities themselves, we use their names and aliases as search queries. We search entity-attribute combinations using the search queries generated by the LLM. We then build another set of queries based on the attributes: First, we determine the natural type for each entity. The natural type of an entity is the super-entity that a human would most likely associate with it, e.g., *bird* for *eagle*, or *clothing* for *hat*. It is neither too general nor too specific, and can typically be used to disambiguate the entity from other entities with the same name. Here, we use an LLM to select the most fitting super-entity from an entity's super-class hierarchy as its natural type. We also let the LLM generate a short description as to why the selected natural type is a good fit for the entity and use it during training as a potential text label. We then replace entity mentions in the attribute search queries with their natural types. For example, the attribute query *eagle in its nest* may turn into *bird in its nest*, or *black BMW M4* into *black car*.

[1] The number of Wikimedia sitelinks is a commonly used and high-quality proxy for the popularity of an entity [26].

[2] In particular, we use three LLMs and merge their generated attributes: Qwen2.5 7B [49], OpenAI GPT-4o, and OpenAI GPT-4o mini (both accessed via the API at platform.openai.com)

3.4. Image search and filtering

We execute our search queries using the image search APIs of Bing and Google. Initial tests on the *living* subset revealed Bing's search results to be of much higher quality at a lower cost, so we rely solely on the Bing API for all other queries. The image search APIs also provide the URL for the website hosting the image, which we use to collect alt texts from the HTML image tag. After downloading images and alt texts, we perform the following postprocessing steps:

- Similar to Changpinyo et al. [4], we apply relaxed filtering heuristics. We do not use any multimodal filtering but instead rely on the search engines to provide image-text correspondences. We remove text that is longer than 500 chars or formatted in JSON. We also remove images with an aspect ratio of more than 4 or covering less than 4096 pixels.
- We deduplicate all downloaded images using the Self-Supervised Descriptor for Image Copy Detection method (SSCD) [27]. For duplicates, we retain the largest image and collect all unique alt texts from the duplicates.
- We detect duplicates between the images and all evaluation datasets using the same SSCD method.

Our final dataset comprises approximately 33M images and 45M alt texts, obtained from 416k queries. This amounts to 79 images per query and 1.4 alt texts per image on average. The total cost for all image search API calls was around 10,000$. See Section 3.4 for an overview over the building blocks and proportions of our dataset.

4. Experimental setup

4.1. Training

We trained all models using the standard CLIP loss [29] with a batch size of 8,192 for pretraining and 32,768 during finetuning, along with random resized crop augmentation. We sampled text labels from both the image alt texts and knowledge graph - for each image, 50% of the time we chose a random alt text for a given image, and 50% of the time we chose randomly between search query, aliases, or descriptions of the corresponding entity. Our training code is based on OpenCLIP [14]. Further training details and an example for the text sampling can be found in the supplementary material. We trained all models for 18 epochs using AdamW [23]. Training on 33M images takes ∼55 hours on 8 L40s GPUs with 48GB VRAM per GPU.

4.2. Evaluated models

On our EntityNet dataset, we trained ViT CLIP models of size B-32 and B-16 from random initialization. For a comparison with a similarly sized dataset, we also trained models on CC12M by downloading all available URLs, and then detecting and removing duplicates relative to the evaluation datasets using the same procedure as detailed in Section 3.4, obtaining 9.3M images. We finetuned B-32 and B-16 CLIP models trained on DataComp-1B on our dataset to compare finetuning and pretraining performance. In addition to the models trained on EntityNet and CC12M, we evaluate the original *OpenAI CLIP* [29] and models pretrained on *DataComp-M/L/1B*, *CommonPool-M/L* [10], and DFN-5B [8]. We also compare with the domain expert model BioCLIP [35], a ViT-B-16 CLIP model finetuned from OpenAI-CLIP on 10M biological image-text pairs.

4.3. Object classification evaluation

To test the VLMs on object classification we use the same procedure as CLIP [29]. Given an image I, class names $C_1, ..., C_N$, image encoder f and text encoder g, we embed the image using the image encoder $\mathbf{v} = f(I)$. To acquire a text embedding for class C_c, the CLIP authors started by directly encoding the class names as $\mathbf{w}_c = g(C_c)$, e.g., *dog*. Alternatively, they created several prompts P using templates, e.g., *graffiti of a dog*, *a photo of the cool dog*, etc., then encoded each prompt, and computed the average embedding: $\mathbf{w}_c = \sum_{p \in P} g(p)/|P|$. They referred to this approach as using "context prompts". Finally, given the image and text embeddings, the prediction p is the class which has the highest cosine similarity to the image. We evaluate all models on just encoding the class name, as well as on using the average embedding of the 80 context prompts that the CLIP authors used for ImageNet, and report the higher top-1 accuracy. For zero-shot object classification, we require models to not have trained on the training set of the benchmark, in order to test "generalization to unseen datasets" [29]

Benchmarks in the generic domain. We evaluate on ImageNet [7], a popular image classification benchmark [32]. We use the ILSVRC2012 validation set, which contains 50,000 images from 1,000 classes. The classes include simple objects such as *park bench*, but also more fine-grained labels like 23 types of terrier dogs, e.g., *Staffordshire Bull Terrier*. We further evaluate the robustness under distribution shifts on ImageNet-A [13], ImageNet-R [12], ImageNet-Sketch [46], ImageNet-V2 [31], and ObjectNet [3] as proposed by Taori et al. [36]. ImageNet-A contains 7500 samples of 200 ImageNet classes. The samples were adversarially filtered to make ResNet-50s misclassify them, providing a more challenging test. ImageNet-R contains 30000 renditions, such as paintings or embroidery, of 200 ImageNet object classes. ImageNet-Sketch contains 50000 sketches, covering 200 ImageNet classes. ImageNet-V2 replicates the original ImageNet generation process, providing an additional 10000 test images. The object-centered ObjectNet contains 18574 images from 113 ImageNet classes with control over background, rotation and viewpoint.

Table 3. Details of our EntityNet dataset. We show the number of unique elements for each column, e.g. the number of images after deduplication or all unique entity aliases in the respective sets.

Query set	Images	Queries	Entities	Aliases	Attributes	Alt texts	Example query
World entity	23M	158k	74k	101k	-	23M	ship
World entity + attribute	19M	139k	6k	16k	20k	16M	small handbag
Living entity	9M	72k	63k	51k	-	8M	kohlrabi
Living entity + attribute	9M	53k	1k	3k	5k	6M	tropical plant
All	33M	416k	135k	149k	23k	45M	-

Benchmarks in the expert domain. We evaluate our models on iNaturalist 2021 [41], a fine-grained species classification benchmark that contains 100k images in the validation set of 10k different species. Similar to Parashar et al. [25], we test models on both the english common name and the latin taxon name. We report the best results over both languages. We further test on the Caltech-UCSD Birds (CUB) [44] dataset, which contains 5,794 images of birds in the original author's test set, each annotated as one of 200 fine-grained bird species, e.g., *grasshopper sparrow*. Additionally, we evaluate on the Rare Species benchmark proposed by Stevens et al. [35], that comprises 400 species with 30 images each specifically tailored to assess generalization to unseen taxa. To comply with the benchmark requirements of not seeing the testing 400 species during training, we exclude all entities and queries from our dataset that appear in RareSpecies, using substring matching. As class names, we evaluate all text types proposed by Stevens et al. [35]: combinations of the latin taxonomy and the english common name. Same as in Section 4.3 we evaluate on both the CLIP ImageNet prompt and no prompt, and report the better of both accuracies.

4.4. Retrieval evaluation

We evaluate the COCO Karpathy test split [15], a subset of 5000 samples from the MS-COCO [19] dataset paired with 5 text each. We also evaluate the 1000 samples in the Karpathy test split of Flickr30k [50] annotated with 5 texts per image, as well as on the 3600 image-text pairs in XM3600 [37]. We report the average of image-to-text and text-to-image recall@1 over all datasets.

5. Results

We evaluate CLIP **pretrained from scratch** on our EntityNet dataset and CLIP models trained on other datasets. In Figures 1 and 3 we contrast the effectiveness of models with their training cost. We show the results in detail in Section 5. In the generic domain, our models surpass others trained on similarly sized datasets while achieving comparable performance on object classification with models trained 20x longer. On image-text retrieval, our model performs similarly to models trained on the same amount of compute. While our pipeline creates a dataset efficient for understanding objects and their properties, understanding complex scenes still requires learning mainly from the alt texts more than from objects and attributes. In the expert domain, we outperform even the largest CLIP models on the challenging iNaturalist 2021 dataset, which requires classifying images among 10k fine-grained species. Our model also excels on CUB by distinguishing 200 bird species bet-

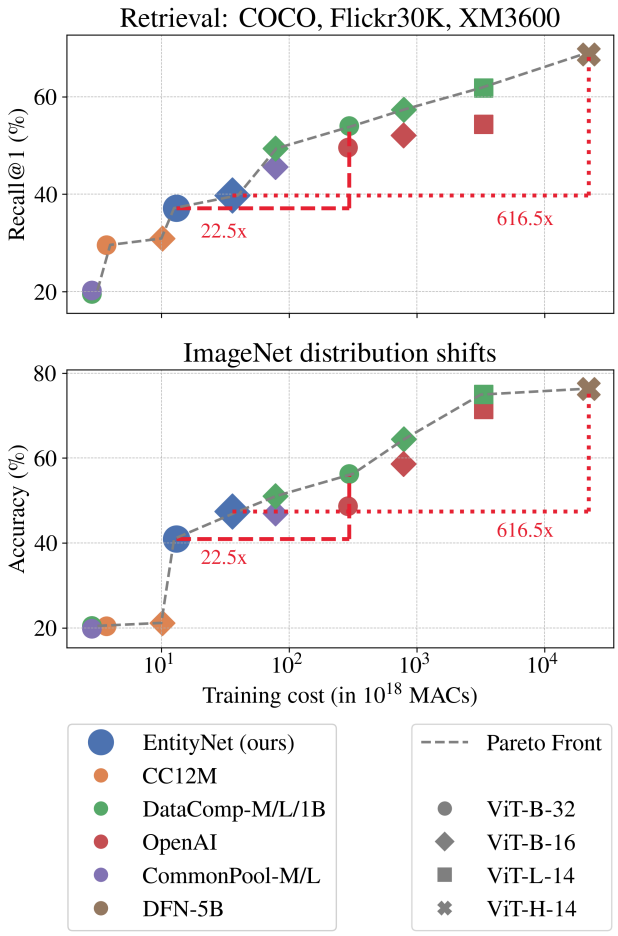


Figure 3. Results on image retrieval and robustness against distribution shifts on ImageNet.

Table 4. Results for training CLIP B-32 and B-16 on our EntityNet dataset from scratch. We mark the **best** and second best result. To measure training cost, we calculate the total MACs (multiply–accumulate operations) performed during training. We only compare zero-shot results and mark results as "–" if the model does not fulfill the zero-shot requirements.

Arch.	Dataset	MACs (1e18)	Images in dataset (M)	Image-Net	Retrie-val	Distr. shift	iNat. 2021	CUB	Rare Species
B-32	CC12M	3.7	9.3	28.6	25.6	18.3	0.7	9.2	–
B-32	CommonPool-M	2.9	128.0	27.2	20.2	19.8	0.8	10.1	–
B-32	DataComp-M	2.9	14.0	29.7	19.5	20.5	1.0	16.8	–
B-32	OpenAI	288.6	400.0	63.4	49.6	48.7	7.4	51.8	–
B-32	DataComp-1B	295.4	1400.0	**69.2**	**54.0**	**56.3**	12.6	73.8	–
B-32	EntityNet (ours)	13.1	32.7	61.5	37.2	41.0	**26.1**	**79.5**	**42.7**
B-16	BioCLIP TreeOfLife-10M	61.3	10.4	18.6	0.8	15.4	–	78.1	38.1
B-16	CommonPool-L	78.2	1280.0	57.8	45.6	47.0	4.1	35.1	–
B-16	DataComp-L	78.2	140.0	63.1	49.4	51.1	6.1	48.1	–
B-16	DataComp-1B	791.4	1400.0	**73.5**	**57.4**	**64.4**	15.3	79.0	–
B-16	OpenAI	784.6	400.0	68.3	52.1	58.6	9.2	56.1	–
B-16	EntityNet (ours)	36.0	32.7	66.2	39.8	47.4	**32.0**	**85.3**	**47.1**
L-14	OpenAI	3328.4	400.0	75.5	54.3	71.4	12.0	62.9	–
L-14	DataComp-1B	3338.6	1400.0	79.2	61.8	74.9	21.1	85.5	–
L-14	DFN-2B	3338.6	2000.0	81.4	64.2	74.8	21.6	86.5	–
H-14	DFN-5B	22164.0	5000.0	**83.4**	**68.7**	**76.3**	25.1	**88.1**	–

Table 5. Results for finetuning the DataComp-1B CLIP model on the EntityNet dataset. We mark the **best** and second best result. To measure training cost, we calculate the total MACs (multiply–accumulate operations) performed during training.

Arch.	Dataset	MACs (1e18)	Images in dataset (M)	Image -Net	Retrie-val	Distr. shift	iNat 2021	CUB
B-32	DataComp-1B (Base model)	295.4	1400.0	69.2	**54.0**	**56.3**	12.6	73.8
B-32	EntityNet	13.3	32.7	**69.5**	50.8	53.3	29.5	83.3
B-32	EntityNet (only living organisms)	4.2	10.2	48.2	31.2	33.3	**37.0**	**87.0**
B-16	DataComp-1B (Base model)	791.4	1400.0	**73.5**	**57.4**	**64.4**	15.3	79.0
B-16	EntityNet	36.1	32.7	**73.5**	52.2	61.0	34.9	86.5
B-16	EntityNet (only living organisms)	11.3	10.2	51.4	34.8	39.2	**42.7**	**90.3**

ter than all other CLIP models of the same size. Further, when compared to the expert model *BioCLIP*, trained specifically for organismal biology at a similar training cost, our model demonstrates superior performance. On the Rare Species benchmark, our model classifies the species unseen during training better than the expert biology model, showing the effectiveness of our dataset collection method over a manually designed living organism training set.

We investigate improving existing CLIP models via **finetuning** in Section 5. The results show that our dataset can be leveraged to create expert CLIP models that outperform both the base model and our model pretrained from scratch on the expert domain. This improvement comes at the cost of trading off some capabilities in the other domains. When finetuning only on the expert domain, we trade off more capabilities, yet obtain even stronger experts.

We further validate and verify our design choices through a **component analysis** in Section 5. Training separately on the generic and the domain expert part of our dataset reveals that, while the best generic model emerges from training on everything, a slightly better expert model is the result of training only on the expert domain (first table segment). However, generalization to unseen species slightly benefits when training on the full dataset, showing that our generic domain data can enhance generalization capabilities within the expert domain. We also observe that generating and downloading attribute queries contributes to improved performance of the pretrained model.

In the second segment of the table, we evaluate the mixture of alt text and knowledge graph labels used during

Table 6. Analysis of performance when varying dataset composition, text sampling and dataset size. We mark the **best** and second best result. To measure training cost, we calculate the total MACs (multiply–accumulate operations) performed during training.

Arch.	Dataset	MACs (1e18)	Images in dataset (M)	Image-Net	Retrie-val	Distr. shift	iNat. 2021	CUB	Rare Species
B-32	Everything	13.1	32.7	**61.5**	**37.2**	**41.0**	26.1	79.5	**42.7**
B-32	No living organisms	9.0	22.5	39.2	32.1	28.0	0.8	6.2	6.9
B-32	Only living organisms	4.1	10.2	36.0	16.5	21.0	**28.6**	**83.2**	42.0
B-32	No attribute queries	8.7	21.8	54.8	28.6	33.8	25.6	79.7	39.2
B-32	50% alt text	13.1	32.7	**61.5**	37.2	**41.0**	**26.1**	**79.5**	**42.7**
B-32	100% alt text	13.1	32.7	59.1	**38.3**	38.1	22.9	78.8	39.7
B-32	0% alt text	13.1	32.7	55.7	13.5	35.5	24.2	78.1	29.7
B-32	Full size	13.1	32.7	**61.5**	**37.2**	**41.0**	**26.1**	**79.5**	**42.7**
B-32	1/2 size	6.6	16.4	54.1	30.3	33.6	20.1	74.1	36.6
B-32	1/4 size	3.3	8.2	45.2	23.5	25.7	13.2	64.9	28.0
B-32	1/8 size	1.6	4.1	33.3	16.6	17.7	7.3	47.8	19.4
B-32	1/16 size	0.8	2.0	19.9	9.4	10.0	2.8	27.4	11.8

training. Notably, both training only on alt texts or only on knowledge graph labels mostly performs worse than our 50-50 mix. The exception is image-text retrieval, where training fully on alt text performs slightly better. Potentially, the knowledge graph labels are less useful for learning the matching between longer text queries and images, and more useful for learning fine-grained object classification.

Finally, we reduce the scale of our dataset by powers of two. While the model performance expectedly drops with reduced dataset size, the efficiency of our dataset per datapoint stays high, with the model still reaching 33% accuracy on ImageNet with only 4M images.

6. Limitations

The proposed data harvesting approach assumes that there is a knowledge graph for the target domain and that there is a searchable database with noisy pairing of images and text. Especially, the latter assumption can induce extra effort in some domains. In the medical domain, for example, there are massive amounts of paired image-text data, but the data is not publicly available and not connected to a regular search engine. The search queries must be adapted appropriately to the respective local database. Another limitation is the small, but significant drop in performance on image-text retrieval and classifying ImageNet distribution shifts in the generic domain, when finetuning a large model with EntityNet. First, our dataset by design has a strong focus on the expert domain and trades off some performance in the generic domain during finetuning. Second, our search pipeline finds many clean object-centric images and annotates them with entity information, which tremendously helps understanding object semantics, but to improve efficiency on image-text retrieval in a similar way, one needs to tackle the quality of alt texts and their alignment to the images [47]. Finally, we focused on searching photos, which explains slightly lower accuracy when classifying paintings and sketches – the EntityNet dataset simply contains a lower percentage of such types of images than datasets like CommonPool.

7. Conclusions

We demonstrated how to use knowledge graphs to harvest datasets that are efficient for training CLIP models. Our strategy allows us to create an expert domain dataset with little manual effort, enabling the development of CLIP models that significantly outperform standard models in the expert domain. The expert domain dataset can be used either for training a model from scratch or for finetuning an existing vanilla model. The substantial size and diversity of the expert domain dataset ensures that the good generalization properties of CLIP exist also in the expert domain, in contrast to training with an over-specialized expert dataset.

Furthermore, we demonstrated that the proposed harvesting strategy is also viable to create a common domain dataset, which allows us to achieve a better quality-compute trade-off than training with previous datasets. Future work can use our EntityNet dataset to train CLIP models with all emergent properties much more efficiently, thus allowing for experiments, where training can be controlled. So far, this has been possible only with models of lacking quality.

References

[1] Encyclopedia of Life, 2018. 3

[2] Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, et al. Phi-3 technical report: A highly capable language model locally on your phone, 2024. 2

[3] Andrei Barbu, David Mayo, Julian Alverio, William Luo, Christopher Wang, Dan Gutfreund, Josh Tenenbaum, and Boris Katz. Objectnet: A large-scale bias-controlled dataset for pushing the limits of object recognition models. *Advances in neural information processing systems*, 32, 2019. 5

[4] 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 *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2021. 3, 5

[5] Jieneng Chen, Qihang Yu, Xiaohui Shen, ALan Yuille, and Liang-Chieh Chen. Design scalable vision models in the vision-language era. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. 3

[6] Xi Chen, Xiao Wang, Soravit Changpinyo, AJ Piergiovanni, Piotr Padlewski, Daniel Salz, Sebastian Goodman, Adam Grycner, Basil Mustafa, Lucas Beyer, Alexander Kolesnikov, Joan Puigcerver, Nan Ding, Keran Rong, Hassan Akbari, Gaurav Mishra, Linting Xue, Ashish V Thapliyal, James Bradbury, Weicheng Kuo, Mojtaba Seyedhosseini, Chao Jia, Burcu Karagol Ayan, Carlos Riquelme Ruiz, Andreas Peter Steiner, Anelia Angelova, Xiaohua Zhai, Neil Houlsby, and Radu Soricut. PaLI: A Jointly-Scaled Multilingual Language-Image Model. In *The Eleventh International Conference on Learning Representations*, 2023. 3

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009. 3, 5

[8] Alex Fang, Albin Madappally Jose, Amit Jain, Ludwig Schmidt, Alexander T Toshev, and Vaishaal Shankar. Data Filtering Networks. In *The Twelfth International Conference on Learning Representations*, 2024. 1, 3, 5

[9] C. Fellbaum. *WordNet: An Electronic Lexical Database*. Mit Press, 1998. 4

[10] Samir Yitzhak Gadre, Gabriel Ilharco, Alex Fang, Jonathan Hayase, Georgios Smyrnis, Thao Nguyen, Ryan Marten, Mitchell Wortsman, Dhruba Ghosh, Jieyu Zhang, Eyal Orgad, Rahim Entezari, Giannis Daras, Sarah Pratt, Vivek Ramanujan, Yonatan Bitton, Kalyani Marathe, Stephen Mussmann, Richard Vencu, Mehdi Cherti, Ranjay Krishna, Pang Wei W Koh, Olga Saukh, Alexander J Ratner, Shuran Song, Hannaneh 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*, pages 27092–27112. Curran Associates, Inc., 2023. 1, 3, 5

[11] Z. Gharaee, Z. Gong, N. Pellegrino, I. Zarubiieva, J. B. Haurum, S. C. Lowe, J. T. A. McKeown, C. Y. Ho, J. McLeod, Y. C. Wei, J. Agda, S. Ratnasingham, D. Steinke, A. X. Chang, G. W. Taylor, and P. Fieguth. A step towards worldwide biodiversity assessment: The BIOSCAN-1M insect dataset. In *Advances in Neural Information Processing Systems*, pages 43593–43619. Curran Associates, Inc., 2023. 3

[12] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8340–8349, 2021. 5

[13] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15262–15271, 2021. 5

[14] 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. 3, 5

[15] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3128–3137, 2015. 6

[16] Runze Li, Dahun Kim, Bir Bhanu, and Weicheng Kuo. RECLIP: Resource-efficient CLIP by Training with Small Images, 2023. 3

[17] Xianhang Li, Zeyu Wang, and Cihang Xie. CLIPA-v2: Scaling CLIP Training with 81.1% Zero-shot ImageNet Accuracy within a $10,000 Budget; An Extra$4,000 Unlocks 81.8% Accuracy. *arXiv preprint arXiv:2306.15658*, 2023. 3, 11

[18] Zichao Li, Cihang Xie, and Ekin Dogus Cubuk. Scaling (Down) CLIP: A Comprehensive Analysis of Data, Architecture, and Training Strategies. *arXiv preprint arXiv:2404.08197*, 2024. 2, 3

[19] 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 *Computer vision–ECCV 2014: 13th European conference, zurich, Switzerland, September 6-12, 2014, proceedings, part v 13*, pages 740–755. Springer, 2014. 6

[20] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved Baselines with Visual Instruction Tuning, 2023. 1

[21] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Advances in Neural Information Processing Systems*, pages 34892–34916. Curran Associates, Inc., 2023.

[22] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. LLaVA-NeXT: Improved reasoning, OCR, and world knowledge, 2024. 1

[23] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. 5

[24] Muhammad Ferjad Naeem, Yongqin Xian, Xiaohua Zhai, Lukas Hoyer, Luc Van Gool, and Federico Tombari. Silc: Improving vision language pretraining with self-distillation. In *European Conference on Computer Vision*, pages 38–55. Springer, 2024. 3

[25] Shubham Parashar, Zhiqiu Lin, Yanan Li, and Shu Kong. Prompting scientific names for zero-shot species recognition. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023. 6

[26] Stanislav Peshterliev, Christophe Dupuy, and Imre Kiss. Self-attention gazetteer embeddings for named-entity recognition. *arXiv preprint arXiv:2004.04060*, 2020. 4

[27] Ed Pizzi, Sreya Dutta Roy, Sugosh Nagavara Ravindra, Priya Goyal, and Matthijs Douze. A Self-Supervised Descriptor for Image Copy Detection. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2022. 5

[28] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis, 2023. 1

[29] 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 38th International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 1, 2, 5

[30] Ahad Rana. Common crawl – building an open web-scale crawl using hadoop, 2010. 3

[31] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *International conference on machine learning*, pages 5389–5400. PMLR, 2019. 5

[32] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 2015. 5

[33] 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. 2

[34] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. LAION-5B: An open large-scale dataset for training next generation image-text models. In *Advances in Neural Information Processing Systems*, pages 25278–25294. Curran Associates, Inc., 2022. 3

[35] Samuel Stevens, Jiaman Wu, Matthew J Thompson, Elizabeth G Campolongo, Chan Hee Song, David Edward Carlyn, Li Dong, Wasila M Dahdul, Charles Stewart, Tanya Berger-Wolf, Wei-Lun Chao, and Yu Su. BioCLIP: A Vision Foundation Model for the Tree of Life. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19412–19424, 2024. 3, 5, 6

[36] Rohan Taori, Achal Dave, Vaishaal Shankar, Nicholas Carlini, Benjamin Recht, and Ludwig Schmidt. Measuring robustness to natural distribution shifts in image classification. *Advances in Neural Information Processing Systems*, 33:18583–18599, 2020. 5

[37] Ashish V Thapliyal, Jordi Pont-Tuset, Xi Chen, and Radu Soricut. Crossmodal-3600: A massively multilingual multimodal evaluation dataset. *arXiv preprint arXiv:2205.12522*, 2022. 6

[38] 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. *Commun. ACM*, 59(2):64–73, 2016. 3

[39] Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, et al. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025. 3

[40] Vishaal Udandarao, Nikhil Parthasarathy, Muhammad Ferjad Naeem, Talfan Evans, Samuel Albanie, Federico Tombari, Yongqin Xian, Alessio Tonioni, and Olivier J Hénaff. Active data curation effectively distills large-scale multimodal models. *arXiv preprint arXiv:2411.18674*, 2024. 3

[41] 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 the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 3, 6

[42] Pavan Kumar Anasosalu Vasu, Hadi Pouransari, Fartash Faghri, Raviteja Vemulapalli, and Oncel Tuzel. MobileCLIP: Fast Image-Text Models through Multi-Modal Reinforced Training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 3

[43] Denny Vrandečić and Markus Krötzsch. Wikidata: A free collaborative knowledge base. *Communications of the ACM*, 57:78–85, 2014. 3

[44] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011. 6

[45] Bo Wan, Michael Tschannen, Yongqin Xian, Filip Pavetic, Ibrahim M Alabdulmohsin, Xiao Wang, André Susano Pinto, Andreas Steiner, Lucas Beyer, and Xiaohua Zhai. Locca: Visual pretraining with location-aware captioners. *Advances in Neural Information Processing Systems*, 37:116355–116387, 2024. 3

[46] Haohan Wang, Songwei Ge, Zachary Lipton, and Eric P Xing. Learning robust global representations by penalizing local predictive power. *Advances in neural information processing systems*, 32, 2019. 5

[47] Hu Xu, Po-Yao Huang, Xiaoqing Ellen Tan, Ching-Feng Yeh, Jacob Kahn, Christine Jou, Gargi Ghosh, Omer Levy, Luke Zettlemoyer, Wen tau Yih, Shang-Wen Li, Saining Xie, and Christoph Feichtenhofer. Altogether: Image captioning via re-aligning alt-text. In *Conference on Empirical Methods in Natural Language Processing*, 2024. 8

[48] Hu Xu, Saining Xie, Xiaoqing Tan, Po-Yao Huang, Russell Howes, Vasu Sharma, Shang-Wen Li, Gargi Ghosh,

Luke Zettlemoyer, and Christoph Feichtenhofer. Demystifying CLIP data. In *The Twelfth International Conference on Learning Representations*, 2024. 3

[49] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024. 4

[50] Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the association for computational linguistics*, 2:67–78, 2014. 6

[51] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid Loss for Language Image Pre-Training. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, 2023. 3

A. Detailed results

We show extensive results on object classification in Section A. For a more detailed analysis of model capabilities on ImageNet, we split the classes into *living* (410 classes) and *other* (590 classes) using WordNet: Since ImageNet labels are built on WordNet nouns, we simply select all labels that are children of the *living things* node for the *living* set. On iNaturalist, in addition to the 2021 version, we also evaluate on the 2019 version which contains 3,030 images in the validation set, each annotated with one of 1,010 species. We test with the same protocol as on iNaturalist 2021, testing on both english and latin class names and reporting the best accuracy. We show additional results on retrieval in Section A and object classification under distribution shift in Section A.

B. Qualitative examples of our dataset

We show randomly sampled images and corresponding textual information of our dataset in Figures 4, 5, 6, and 7. We show an example of our text label sampling strategy in Section B.

C. Hyperparameter settings

We report hyperparameters for our experiments in Section E. Similar to Li et al. Li et al. [17], we reduce the context size of the text encoder down from 77 to 32 to reduce VRAM and training time requirements. For a fair comparison with other CLIP models, we report all training cost and training duration as if the training was run at a context length of 77.

D. Image Search APIs

Google The Google Image Search API is available via the Google Cloud Platform, and requires an existing programmable search engine to function. It returns up to 10 images per request and page with a limit of 10 pages, i.e., 100 images per query. It costs 5$ per 1,000 API calls, resulting in costs of about 500$ to download 1M images. We found the search results from the Google API to be quite different, and arguably worse, from the ones returned when using the regular Google image search. For all our API requests we set the parameter *imgColorType* to *color*, *imgType* to *photo*, *lr* to *lang_en*, and *excludeTerms* to *drawing clipart illustration cartoon vector painting*. This way we get mostly real-world images in our search results. We additionally add all aliases and the natural type of the sought entity to the *orTerms* parameter for entity and entity-attribute queries. Because the Google API returns only up to 10 images per request and page, we search for the following number of pages: 2 pages each for entity queries, 4 pages each for entity-attribute queries, and 10 pages each for all natural-type-attribute queries. We started our search with queries from the *living* subset on both the Google and Bing APIs. We found the quality and value-for-money ratio of the Bing API to be better, and therefore switched to only using Bing for all other queries.

Bing The Bing Image Search API is available via Microsoft Azure. It returns up to 150 images per request and has no restrictions on the number of accessible pages. It costs 18$ per 1,000 API calls, resulting in costs of about 120$ to download 1M images. In our experience, the returned images closely match the ones from the regular Bing image search. For all our API requests we set the parameter *imageType* to *Photo* and *color* to *ColorOnly*. Unlike the Google API, Bing does not have a way to specify *orTerms* via a separate request parameter, so we add the natural type of the sought entity to the search query directly, e.g., we search for *jaguar animal*. The Bing API returns 150 images per request and page, we request one page for each query.

E. Querying entities with SPARQL

The SPARQL query over Wikidata used to harvest all entities under a super-entity is displayed in Figure 8. It returns a list of entities from a specified target domain as defined by one or more super-entities. The super-entities can be determined manually by searching for appropriate entities on the Wikidata website. For example, if we want to build a dataset about vehicles, we can set the super-entity to vehicle (Q42889), as done in Section E. We list the super-entities we considered for our dataset and the relevant statistics in Section E for included and Section E for excluded super-entities.

Table 7. Detailed object classification results. The table is grouped into training from scratch, finetuning, and analyzing components, the groups are separated by double horizontal lines. Each part is again split by single horizontal lines into groups of same model size or same component analysis. Models marked with ⋆ are finetuned. We mark the **best** and second best result. To measure training cost, we calculate the total MACs (multiply–accumulate operations) performed during training. We only compare zero-shot results and mark results as "–" if the model does not fulfill the zero-shot requirements.

Arch.		Dataset	MACs (1e18)	Images in dataset	ImageNet 1k	ImageNet Living	ImageNet other	iNaturalist 2019	iNaturalist 2021	CUB	Rare species
				# Classes →	1,000	410	590	1,010	10k	200	400
B-32		CC12M	3.7	9.3	28.6	27.6	31.1	2.0	0.7	9.2	–
B-32		CommonPool-M	2.9	128.0	27.2	20.6	33.5	2.6	0.8	10.1	–
B-32		DataComp-M	2.9	14.0	29.7	25.5	34.5	3.0	1.0	16.8	–
B-32		OpenAI	288.6	400.0	63.4	65.5	63.1	10.9	7.4	51.8	–
B-32		DataComp-1B	295.4	1400.0	**69.2**	**71.2**	**69.1**	16.7	12.6	73.8	–
B-32		EntityNet	13.1	32.7	61.5	68.5	57.9	**38.3**	**26.1**	**79.5**	**42.7**
B-16	⋆	BioCLIP TreeOfLife-10M	61.3	10.4	18.6	44.3	2.6	–	–	78.1	38.1
B-16		CommonPool-L	78.2	1280.0	57.8	53.2	62.4	6.9	4.1	35.1	–
B-16		DataComp-L	78.2	140.0	63.1	61.8	65.3	9.1	6.1	48.1	–
B-16		DataComp-1B	791.4	1400.0	**73.5**	**75.9**	**73.2**	19.5	15.3	79.0	–
B-16		OpenAI	784.6	400.0	68.3	71.5	67.4	12.5	9.2	56.1	–
B-16		EntityNet	36.0	32.7	66.2	73.9	62.0	**42.2**	**32.0**	**85.3**	**47.1**
L-14		OpenAI	3328.4	400.0	75.5	78.9	74.5	15.2	12.0	62.9	–
L-14		DataComp-1B	3338.6	1400.0	79.2	82.1	78.4	23.6	21.1	85.5	–
L-14		DFN-2B	3338.6	2000.0	81.4	83.7	80.9	24.1	21.6	86.5	–
H-14		DFN-5B	22164.0	5000.0	**83.4**	**85.4**	**83.2**	31.4	25.1	**88.1**	–
B-32		DataComp-1B (Base model)	295.4	1400.0	69.2	71.2	**69.1**	16.7	12.6	73.8	–
B-32	⋆	EntityNet	13.3	32.7	**69.5**	73.6	67.8	41.5	29.5	83.3	–
B-32	⋆	EntityNet (only living organisms)	4.2	10.2	48.2	**76.3**	30.9	**49.3**	**37.0**	**87.0**	–
B-16		DataComp-1B (Base model)	791.4	1400.0	**73.5**	75.9	**73.2**	19.5	15.3	79.0	–
B-16	⋆	EntityNet	36.1	32.7	**73.5**	78.1	71.5	46.7	34.9	86.5	–
B-16	⋆	EntityNet (only living organisms)	11.3	10.2	51.4	**80.2**	33.8	**54.3**	**42.7**	**90.3**	–
B-32		EntityNet Everything	13.1	32.7	**61.5**	**68.5**	**57.9**	38.3	26.1	79.5	**42.7**
B-32		EntityNet (no living organisms)	9.0	22.5	39.2	17.5	56.1	1.7	0.8	6.2	6.9
B-32		EntityNet (only living organisms)	4.1	10.2	36.0	**68.5**	15.4	**41.4**	**28.6**	**83.2**	42.0
B-32		EntityNet (no attribute queries)	8.7	21.8	54.8	63.0	50.4	36.4	25.6	79.7	39.2
B-32		EntityNet 50% alt text	13.1	32.7	**61.5**	**68.5**	**57.9**	**38.3**	**26.1**	**79.5**	**42.7**
B-32		EntityNet 100% alt text	13.1	32.7	59.1	66.5	55.4	36.4	22.9	78.8	39.7
B-32		EntityNet 0% alt text	13.1	32.7	55.7	64.4	51.5	35.4	24.2	78.1	29.7
B-32		EntityNet Full size	13.1	32.7	**61.5**	**68.5**	**57.9**	**38.3**	**26.1**	**79.5**	**42.7**
B-32		EntityNet 1/2 size	6.6	16.4	54.1	61.5	50.5	30.7	20.1	74.1	36.6
B-32		EntityNet 1/4 size	3.3	8.2	45.2	52.4	41.8	23.4	13.2	64.9	28.0
B-32		EntityNet 1/8 size	1.6	4.1	33.3	39.6	30.5	14.5	7.3	47.8	19.4
B-32		EntityNet 1/16 size	0.8	2.0	19.9	25.0	18.1	7.6	2.8	27.4	11.8
B-32		EntityNet batch size 2,048	13.1	32.7	59.7	65.4	57.2	31.5	21.0	74.1	38.7
B-32		EntityNet batch size 4,096	13.1	32.7	60.9	67.0	**58.0**	36.2	24.1	77.6	41.0
B-32		EntityNet batch size 8,192	13.1	32.7	**61.5**	**68.5**	57.9	38.3	26.1	79.5	**42.7**
B-32		EntityNet batch size 16,384	13.2	32.7	60.5	68.2	56.6	38.8	**26.9**	81.2	42.1
B-32		EntityNet batch size 32,768	13.3	32.7	58.6	67.0	54.2	**39.5**	26.3	**81.3**	41.7

Table 8. Detailed retrieval results. The table is grouped into training from scratch, finetuning, and analyzing components, the groups are separated by double horizontal lines. Each part is again split by single horizontal lines into groups of same model size or same component analysis. Models marked with ⋆ are finetuned. We mark the **best** and second best result. To measure training cost, we calculate the total MACs (multiply–accumulate operations) performed during training.

Arch.		Dataset	MACs (1e18)	Images in dataset	Retrieval Average	COCO test I2T	COCO test T2I	F30K test I2T	F30K test T2I	XM3600 test I2T	XM3600 test T2I
B-32		CC12M	3.7	9.3	25.6	22.4	15.2	37.2	27.1	26.1	25.5
B-32		CommonPool-M	2.9	128.0	20.2	18.3	11.2	29.9	18.9	23.6	19.6
B-32		DataComp-M	2.9	14.0	19.5	17.1	11.0	26.0	18.0	23.6	21.5
B-32		OpenAI	288.6	400.0	49.6	50.1	30.5	77.5	58.8	43.4	37.2
B-32		DataComp-1B	295.4	1400.0	**54.0**	**53.5**	**37.1**	**78.8**	**61.1**	**48.3**	**45.3**
B-32		EntityNet	13.1	32.7	37.2	32.8	22.2	52.3	37.8	40.6	37.3
B-16	⋆	BioCLIP TreeOfLife-10M	61.3	10.4	0.8	0.4	0.2	0.9	0.6	1.8	1.1
B-16		CommonPool-L	78.2	1280.0	45.6	44.4	28.8	68.3	51.0	42.1	39.2
B-16		DataComp-L	78.2	140.0	49.4	48.7	32.2	73.5	55.1	44.7	42.1
B-16		DataComp-1B	791.4	1400.0	**57.4**	**57.5**	**40.2**	**84.9**	**67.3**	**47.9**	**46.5**
B-16		OpenAI	784.6	400.0	52.1	52.5	33.1	81.9	62.0	43.8	39.3
B-16		EntityNet	36.0	32.7	39.8	36.0	25.2	57.1	43.3	39.9	37.1
L-14		OpenAI	3328.4	400.0	54.3	56.3	36.5	85.1	65.2	44.5	38.4
L-14		DataComp-1B	3338.6	1400.0	61.8	63.2	45.8	89.5	73.4	50.4	48.6
L-14		DFN-2B	3338.6	2000.0	64.2	65.7	48.6	89.6	75.3	53.6	52.7
H-14		DFN-5B	22164.0	5000.0	**68.7**	**72.3**	**53.9**	**93.0**	**80.2**	**57.6**	**55.4**
B-32		DataComp-1B (Base model)	295.4	1400.0	**54.0**	**53.5**	**37.1**	**78.8**	**61.1**	**48.3**	45.3
B-32	⋆	EntityNet	13.3	32.7	50.8	48.1	34.4	72.1	57.1	47.8	**45.6**
B-32	⋆	EntityNet (only living organisms)	4.2	10.2	31.2	28.0	19.7	44.8	33.7	30.9	29.9
B-16		DataComp-1B (Base model)	791.4	1400.0	**57.4**	**57.5**	**40.2**	**84.9**	**67.3**	**47.9**	**46.5**
B-16	⋆	EntityNet	36.1	32.7	52.2	50.4	36.4	75.6	59.4	47.3	44.3
B-16	⋆	EntityNet (only living organisms)	11.3	10.2	34.8	31.8	22.8	51.7	37.1	33.5	32.0
B-32		EntityNet Everything	13.1	32.7	**37.2**	**32.8**	**22.2**	**52.3**	**37.8**	**40.6**	**37.3**
B-32		EntityNet (no living organisms)	9.0	22.5	32.1	28.6	18.7	46.3	33.6	33.2	32.3
B-32		EntityNet (only living organisms)	4.1	10.2	16.5	12.8	9.8	23.4	15.3	19.0	18.6
B-32		EntityNet (no attribute queries)	8.7	21.8	28.6	23.6	16.1	41.3	27.3	32.3	30.8
B-32		EntityNet 50% alt text	13.1	32.7	37.2	32.8	22.2	52.3	37.8	**40.6**	37.3
B-32		EntityNet 100% alt text	13.1	32.7	**38.3**	**35.2**	**23.2**	**53.4**	**39.8**	40.5	**38.0**
B-32		EntityNet 0% alt text	13.1	32.7	13.5	8.7	6.1	19.2	11.7	18.3	17.0
B-32		EntityNet Full size	13.1	32.7	**37.2**	**32.8**	**22.2**	**52.3**	**37.8**	**40.6**	**37.3**
B-32		EntityNet 1/2 size	6.6	16.4	30.3	27.0	17.6	42.7	29.7	33.3	31.6
B-32		EntityNet 1/4 size	3.3	8.2	23.5	19.5	13.0	31.1	22.9	27.6	27.2
B-32		EntityNet 1/8 size	1.6	4.1	16.6	12.8	8.7	20.7	14.2	21.9	20.9
B-32		EntityNet 1/16 size	0.8	2.0	9.4	7.1	5.0	10.0	7.1	13.9	13.5
B-32		EntityNet batch size 2,048	13.1	32.7	35.6	30.8	20.9	50.2	37.2	38.2	36.4
B-32		EntityNet batch size 4,096	13.1	32.7	36.4	31.6	22.1	51.7	37.0	39.1	37.2
B-32		EntityNet batch size 8,192	13.1	32.7	**37.2**	**32.8**	**22.2**	**52.3**	**37.8**	**40.6**	**37.3**
B-32		EntityNet batch size 16,384	13.2	32.7	35.8	32.2	21.8	50.5	36.9	37.2	36.1
B-32		EntityNet batch size 32,768	13.3	32.7	34.5	31.0	20.8	48.9	34.5	36.2	35.3

Table 9. Detailed results on ImageNet distribution shifts. The table is grouped into training from scratch, finetuning, and analyzing components, the groups are separated by double horizontal lines. Each part is again split by single horizontal lines into groups of same model size or same component analysis. Models marked with ⋆ are finetuned. We mark the **best** and second best result. To measure training cost, we calculate the total MACs (multiply–accumulate operations) performed during training. *INet:* ImageNet.

Arch.		Dataset	MACs (1e18)	Images in dataset (M)	INet 1K	Ave-rage	INet V2	INet R	Object-Net	INet Sketch	INet A
		# Classes →			1,000	–	1000	200	1000	1000	200
B-32		CC12M	3.7	9.3	28.6	18.3	24.2	34.5	12.1	16.0	4.7
B-32		CommonPool-M	2.9	128.0	27.2	19.8	22.5	33.0	20.9	18.4	4.3
B-32		DataComp-M	2.9	14.0	29.7	20.5	24.4	34.0	19.7	19.3	4.9
B-32		OpenAI	288.6	400.0	63.4	48.7	56.0	69.4	44.2	42.3	**31.5**
B-32		DataComp-1B	295.4	1400.0	**69.2**	**56.3**	**60.8**	**78.2**	**55.2**	**56.8**	30.5
B-32		EntityNet	13.1	32.7	61.5	41.0	53.6	58.8	32.6	45.0	14.9
B-16	⋆	BioCLIP TreeOfLife-10M	61.3	10.4	18.6	15.4	17.7	16.0	3.2	7.3	32.9
B-16		CommonPool-L	78.2	1280.0	57.8	47.0	50.0	68.4	49.1	45.9	21.7
B-16		DataComp-L	78.2	140.0	63.1	51.1	55.2	71.8	53.1	49.7	25.5
B-16		DataComp-1B	791.4	1400.0	**73.5**	**64.4**	**66.0**	**83.6**	**63.7**	**60.4**	48.4
B-16		OpenAI	784.6	400.0	68.3	58.6	61.9	77.7	55.3	48.2	**49.9**
B-16		EntityNet	36.0	32.7	66.2	47.4	59.2	64.1	40.9	48.9	23.9
L-14		OpenAI	3328.4	400.0	75.5	71.4	69.9	87.9	69.0	59.6	**70.7**
L-14		DataComp-1B	3338.6	1400.0	79.2	74.9	72.0	90.8	**74.3**	68.0	69.6
L-14		DFN-2B	3338.6	2000.0	81.4	74.8	74.6	90.0	74.1	68.3	66.8
H-14		DFN-5B	22164.0	5000.0	**83.4**	**76.3**	**77.4**	**93.0**	68.4	**72.8**	69.9
B-32		DataComp-1B (Base model)	295.4	1400.0	69.2	**56.3**	60.8	**78.2**	**55.2**	**56.8**	**30.5**
B-32	⋆	EntityNet	13.3	32.7	**69.5**	53.3	**61.9**	74.2	47.9	**56.8**	25.6
B-32	⋆	EntityNet (only living organisms)	4.2	10.2	48.2	33.3	43.1	56.7	19.1	32.1	15.8
B-16		DataComp-1B (Base model)	791.4	1400.0	**73.5**	**64.4**	66.0	**83.6**	**63.7**	**60.4**	**48.4**
B-16	⋆	EntityNet	36.1	32.7	**73.5**	61.0	**66.5**	79.0	56.6	59.8	42.9
B-16	⋆	EntityNet (only living organisms)	11.3	10.2	51.4	39.2	46.1	62.1	25.9	35.1	26.8
B-32		EntityNet Everything	13.1	32.7	**61.5**	**41.0**	**53.6**	**58.8**	**32.6**	**45.0**	**14.9**
B-32		EntityNet (no living organisms)	9.0	22.5	39.2	28.0	33.6	37.7	29.1	32.0	7.5
B-32		EntityNet (only living organisms)	4.1	10.2	36.0	21.0	31.5	39.6	8.1	17.6	8.1
B-32		EntityNet (no attribute queries)	8.7	21.8	54.8	33.8	47.9	49.2	24.4	36.9	10.5
B-32		EntityNet 50% alt text	13.1	32.7	**61.5**	**41.0**	**53.6**	**58.8**	**32.6**	**45.0**	**14.9**
B-32		EntityNet 100% alt text	13.1	32.7	59.1	38.1	51.1	55.6	30.1	40.8	13.1
B-32		EntityNet 0% alt text	13.1	32.7	55.7	35.5	48.0	53.0	27.1	36.8	12.4
B-32		EntityNet Full size	13.1	32.7	**61.5**	**41.0**	**53.6**	**58.8**	**32.6**	**45.0**	**14.9**
B-32		EntityNet 1/2 size	6.6	16.4	54.1	33.6	47.2	51.0	24.4	36.5	8.9
B-32		EntityNet 1/4 size	3.3	8.2	45.2	25.7	39.1	39.9	18.1	25.6	5.9
B-32		EntityNet 1/8 size	1.6	4.1	33.3	17.7	28.5	28.7	12.0	15.3	4.2
B-32		EntityNet 1/16 size	0.8	2.0	19.9	10.0	17.2	17.6	6.8	5.7	2.8
B-32		EntityNet batch size 2,048	13.1	32.7	59.7	40.4	52.5	59.4	31.3	44.5	14.4
B-32		EntityNet batch size 4,096	13.1	32.7	60.9	**41.0**	53.0	**60.0**	32.1	44.9	**14.9**
B-32		EntityNet batch size 8,192	13.1	32.7	**61.5**	**41.0**	**53.6**	58.8	**32.6**	**45.0**	**14.9**
B-32		EntityNet batch size 16,384	13.2	32.7	60.5	39.7	53.2	58.0	30.6	42.9	13.8
B-32		EntityNet batch size 32,768	13.3	32.7	58.6	37.2	51.3	54.6	27.9	40.6	11.6

Entity + *attribute*
running polecat

A wonderful polecat in its woodland surroundings

Polecats Unveiled: Sleek Predators in the Countryside (Mustela Putorius) - Glenlivet Wildlife

Polecats Unveiled: Sleek Predators in the Countryside (Mustela Putorius)

Black Polecat Photos and Premium High Res Pictures - Getty Images

Do Cats Eat Ferrets – What You Should Know! – FAQcats.com

Do Cats Eat Ferrets – What You Should Know!

Entity + *attribute*
summer Canada goose

Canada Geese Goose Branta - Free photo on Pixabay - Pixabay

Canada Geese Goose Branta · Free photo on Pixabay

Facts about geese

Canada Geese Goose Branta Free Photo On Pixabay Pixabay, 45% OFF

Entity + *attribute*
wild tortoise

Greek Tortoise Testudo Graeca Hiding Shell Stock Photo 1425661328 | Shutterstock

Elongated Tortoise Indotestudo Elongata Yellow Tortoise Stock Photo 1463951543 | Shutterstock

Entity
Orbea decaisneana

Orbea decaisneana subs. hesperidum f. cristata

Entity + *attribute*
old walrus

What Is A Walrus?

What is a Walrus - Walrus Habitat and Behavior - Wild Focus Expeditions

Portrait of an old bull walrus resting on his teeth, tooth walker

Figure 4. Randomly sampled images from the EntityNet *living* subset together with query type, search query, and alt texts (separated by new lines).

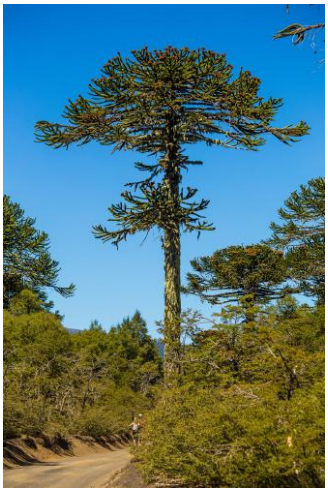
	*Entity* Junín red squirrel	Curious Eurasian Red Squirrel, Sciurus Vulgaris, Running and Jumping . . .
	*Entity* chile pine	Araucaria araucana - Wikipedia Araucaria araucana - Wikipedia, la enciclopedia libre Araucaria araucana - Wikipedia | Trees to plant, Denver botanic gardens ... a tall tree in the middle of a forest Araucaria araucana - Wikipedia | Denver botanic gardens, Outdoor plants ... Monkey Puzzle Plant main Monkey Puzzle Plant Care & Growing Basics: Water, Light, Soil, Propagation etc.
	*Entity* Cedronella canariensis	Cedronella canariensis
	*Entity* Barbuda Warbler	In the face of elite tourism projects, the Barbuda Warbler isn't the only one that might lose its home Barbuda Warbler - Setophaga subita - Birds of the World / - Barbuda Warbler
	*Natural type + attribute* long neck animal	The Strange Elegance of the Giraffe-Necked Antelope | The Ark In Space

Figure 5. Randomly sampled images from the EntityNet *living* subset together with query type, search query, and alt texts (separated by new lines).

	*Natural type + attribute* carpaccio with lemon	Tuna Carpaccio with Fennel and Lemon Recipe - Great British Chefs Tuna carpaccio with fennel and lemon
	*Natural type + attribute* index finger nail	A Macro Of Index Finger Nail Stock Footage SBV-301021776 - Storyblocks
	*Entity* akaogiite	Akaogiite Cut Out Stock Images
	*Entity* public bookcase	manufacta est: public bookcase
	*Entity* BK 117	Eurocopter-Kawasaki BK-117B-2 - DRF - Deutsche Rettungsflugwacht | Aviation Photo #1053819 | Airliners.net Eurocopter-Kawasaki BK-117B-2 - DRF - Deutsche Rettungsflugwacht

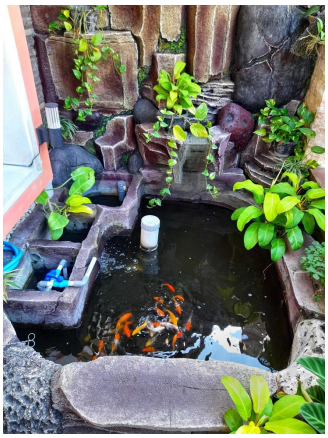
Figure 6. Randomly sampled images from the EntityNet *non-living* subset together with query type, search query, and alt texts (separated by new lines).

Entity
cowboy boots

Ariat Heritage Rough Stock Cowboy Boots - Square Toe - Country Outfitter

Entity + *attribute*
nacre jewelry

Genuine Nacre Necklace Pearl Jewelry Mother Of Pearls Beads | Etsy



Entity + *attribute*
small artificial pond

Small backyard pond decoration. Artificial pond in garden. Pool aquatic plants. Pond border decoration. photo

Small backyard pond decoration. Artificial pond in garden. Pool aquatic plants. Pond border decoration.

Small backyard pond decoration. Artificial pond in garden. Pool aquatic plants. Pond border decoration. 9562060 Stock Photo at Vecteezy

Small backyard pond decoration. Artificial pond in garden. Pool aquatic plants. Pond border decoration. Free Photo

Entity + *attribute*
berimbau at rest

Premium AI Image | Best Berimbau With Handle Isolated On White Background

Entity + *attribute*
assembled motherboard

Red Team rocking: Build the ultimate AMD gaming PC | PCWorld

amd fx cpu

Figure 7. Randomly sampled images from the EntityNet *non-living* subset together with query type, search query, and alt texts (separated by new lines).

Table 10. Example of our text label sampling strategy for an image returned from the entity query of zipper. Probability mass is split 50/50 between image alt texts and texts from the knowledge graph. Between alt texts, we chose uniformly. Between knowledge graph texts we chose the search query 25% of the time, a description 10% of the time (uniformly between descriptions), and an alias otherwise (uniformly between all aliases).

Text	Chance	Source
Zipper PNG	25%	Alt text
yellow zipper PNG image	25%	Alt text
zipper	12.5%	Search query
zip	5.5%	Alias
dingy	5.5%	Alias
clasp locker	5.5%	Alias
fly	5.5%	Alias
zip fastener	5.5%	Alias
device for fastening the edges of an opening of fabric or other flexible material	2.5%	Description
A device used for fastening, typically made of physical material.	2.5%	Description

Table 11. Hyperparameters used for training and finetuning.

Dataset	Model	Batch Size	Max LR	Weight Decay	Epochs	Warmup (epochs)	eps	Beta 1	Beta 2
CC12M	ViT-B/32	8k	5e-4	0.2	18	2	1e-8	0.9	0.98
CC12M	ViT-B/16	8k	5e-4	0.2	18	2	1e-8	0.9	0.98
Ours	ViT-B/32	8k	5e-4	0.2	18	2	1e-8	0.9	0.98
Ours	ViT-B/16	8k	5e-4	0.2	18	2	1e-8	0.9	0.98
Ours, finetuning	ViT-B/32	32k	5e-5	0.2	18	2	1e-8	0.9	0.98
Ours, finetuning	ViT-B/16	32k	5e-5	0.2	18	2	1e-8	0.9	0.98

```
PREFIX wdt: <http://www.wikidata.org/prop/direct/>
PREFIX wd: <http://www.wikidata.org/entity/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX schema: <http://schema.org/>
PREFIX wikibase: <http://wikiba.se/ontology#>
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
SELECT DISTINCT
  ?ent
  ?label
  ?desc
  ?links
  (GROUP_CONCAT(DISTINCT ?alias; SEPARATOR=";;;") AS ?aliases)
WHERE {
  VALUES ?typ { wd:Q42889 }
  ?ent wdt:P279* ?typ .
  ?ent rdfs:label ?label .
  FILTER(LANG(?label) = "en")
  ?ent ^schema:about/wikibase:sitelinks ?links .
  FILTER(?links >= 5)
  OPTIONAL {
    ?ent schema:description ?desc .
    FILTER(LANG(?desc) = "en")
  }
  OPTIONAL {
    ?ent skos:altLabel ?alias .
    FILTER(LANG(?alias) = "en")
  }
}
GROUP BY ?ent ?label ?desc ?links
ORDER BY DESC(?links)
```

Figure 8. Generic SPARQL query for extracting entities from Wikidata that are related to a given set of super-entities. The super-entities are manually set within the *VALUES ?typ* { ... } clause. In this example it is the motor car entity wd:Q42889. A minimum number of sitelinks can also be specified to filter out unpopular entities, here it is set to 5.

Table 12. Vehicle entities and accompanying information as extracted from the Wikidata knowledge graph. Showing the first 5 and last 5 out of 17,015 entities. Note that we only collect entities with sitelinks ≥ 5. The corresponding SPARQL query is shown in Figure 8.

Identifier	Name	Description	Sitelinks	Aliases
Q1420	motor car	motorized road vehicle designed to carry one to eight people rather than primarily goods	237	auto / motor vehicle / motor cars / motorcar / cars / car / automobiles / automobile / autocar
Q11442	bicycle	pedal-driven two-wheel vehicle	203	bike / Bicycles / cycle / pushbike / pedal cycle / pedal bike
Q197	airplane	powered fixed-wing aircraft	196	airplane, aeroplane, plane / powered fixed-wing aircraft / planes / plane / aeroplane / fixed-wing powered aircraft / fixed-wing airplane / aeroplanes / fixed-wing aeroplane / airplanes
Q870	train	form of rail transport consisting of a series of connected vehicles	193	rail-train / trains / railway train / railtrain / rail train / railroad train
Q11446	ship	large buoyant watercraft	178	marine vessel / vessel / water vessel / ships
Q813876	Bedford JJL	motor vehicle	5	
Q7077241	Odakyu 20000 series RSE	Japanese electric multiple unit trainset	5	RSE / Romancecar RSE / Resort Super Express / Odakyu Romancecar RSE / 20000 series
Q812263	Bavarian Pt 2/3	class of 97 German 2-4-0T locomotives	5	ÖBB 770 / DR Class 70.0 / DRG Class 70.0
Q9177196	Bombardier CRJ1000	regional jet airliner	5	CRJ1000
Q812260	Bavarian PtL 2/2	class of 6+29+13 German 0-4-0T locomotives	5	DB Class 98.3 / DRG Class 98.3 / ÖBB 688

Table 13. The super-entities for building our EntityNet dataset to describe the visual world. The *aliases* column refers to the set of all aliases collected from the entities. The numbers in this table are slightly higher than the ones we report in the main paper, because they refer to the raw counts of entities and aliases before profanity filtering and the removal of entities that return no results in the image search.

Super-entity	Description	Examples	Entities	Aliases
product	Anything that can be offered to a market	banh mi, navigation system, PlayStation 2	63,676	144,715
substance	Any composed matter whose origin is either biological, chemical, or mineral	solid lubricant, Chinese tea, eye cups	34,259	111,383
physical tool	Physical item that can be used to achieve a goal	Patient lift, police transport, instant camera	32,727	71,227
animal	Kingdom of multicellular eukaryotic organisms	saw-scaled viper, Sporathraupis cyanocephala, Rufous mouse-eared bat	28,000	76,408
plant	Living thing in the kingdom of photosynthetic eukaryotes	Whitebark Pine, Eucalyptus coccifera, wig knapweed	28,000	55,925
material	Substance that can occur in different amounts, all with some similar [mixture of some] characteristics, and with which objects can be made	dietary proteins, stone slab tomb, safflower oil	18,021	40,822
vehicle	Mobile machine used for transport, whether it has an engine or not, including wheeled and tracked vehicles, air-, water-, and space-craft	shipwrecks (objects), Evergreen A-class container ship, VTOL aircraft	17,015	37,849
geographical feature	Components of planets that can be geographically located	hydrothermal Vents, grooves, street lamp	8,683	19,030
food	Any substance consumed to provide nutritional support for the body; form of energy stored in chemical	coffee milk, tikka, Friesian Clove	8,464	15,332
architectural structure	Human-designed and -made structure	rock temples, summerhouse, house of worship	4,507	10,354
anatomical structure	Entity with a single connected inherent 3d shape that's created by coordinated expression of the organism's own dna	bronchi, maxillary wisdom tooth, turtle shell	4,394	9,999
facility	Place, equipment, or service to support a specific function	public toilet, automobile servicing shop, industrial park	2,767	6,740
physical activity	Human physical activity consisting of voluntary bodily movement by skeletal muscles	American rules football, archery, water-skiing	2,228	4,422
clothing	Covering worn on the body	blucher shoe, G-suit, one-piece swimsuit	1,929	4,313
building	Structure, typically with a roof and walls, standing more or less permanently in one place	shoestore, family restaurant, factory outlet	1,655	3,964
musical instrument	Device created or adapted to make musical sounds	electroencephalophone, Chinese flutes, oboe	1,450	3,493
organ	Collection of tissues with similar functions	nasal bone, cranial nerves, ulnar collateral ligament of elbow	1,155	2,450
furniture	Movable objects used to equip households, offices, or shops for purposes such as storage, seating, sleeping	faldstool, airline seat, bicycle parking rack	388	933
body of water	Any significant accumulation of water, generally on a planet's surface	dammed lake, deep-sea hydrothermal vent, marshland	379	792
weather	State of the atmosphere	cold snap, tropical cyclone, sea of fog	151	304
precipitation	Liquid or solid water that falls to the ground	hail, thunderstorm, snowfall	43	72
Total	Before deduplication		259,891	620,527
Total	After deduplication		146,985	368,062
Total	After deduplication, without animals and plants		90,985	235,795

Table 14. We consider these super-entities either non-visual, irrelevant, or too specific and do not select related entities when building our dataset.

Super-entity	Description
abstract entity	entity that does not have a physical existence, including abstract objects and properties
astronomical object	physical body of astronomically-significant size, mass, or role, naturally occurring in a universe
city	large human settlement
concept	semantic unit understood in different ways, e.g. as mental representation, ability or abstract object (philosophy)
continent	large landmass identified by convention
country	distinct territorial body or political entity
historical event	particular incident in history that brings about a historical change
history	past events and their tracks or records
imaginary character	character known only from narrations (fictional or in a factual manner) without a proof of existence; includes fictional, mythical, legendary or religious characters and similar
language	particular system of communication, often named for the region or peoples that use it
language	structured system of communication
medical procedure	process of medicine done to heal; course of action intended to achieve a result in the delivery of healthcare
organization	social entity established to meet needs or pursue goals
planet	celestial body directly orbiting a star or stellar remnant
religion	social-cultural system
representation	entity or process that portrays something else, usually in a simplified or approximated manner
role	social role with a set of powers and responsibilities within an organization
science	systematic endeavor that builds and organizes knowledge, and the set of knowledge produced by this system
social system	patterned series of interrelationships existing between individuals, groups, and institutions
speciality	field limited to a specific area of knowledge; specialization in an occupation or branch of learning; a specific use
star	astronomical object consisting of a luminous spheroid of plasma held together by its own gravity
temporal entity	thing that can be contained within a period of time, or change in state (e.g. events, periods, acts)
work of art	aesthetic item or artistic creation; object whose value is its beauty only, not practical usefulness
written work	any work expressed in writing, such as inscriptions, manuscripts, documents or maps