Open-domain Visual Entity Recognition: Towards Recognizing Millions of Wikipedia Entities

Hexiang Hu♠ Yi Luan♠ Yang Chen♠♡† Urvashi Khandelwal♠
Mandar Joshi♠ Kenton Lee♠ Kristina Toutanova♠ Ming-Wei Chang♠
♠ Google Deepmind ♡ Georgia Institute of Technology

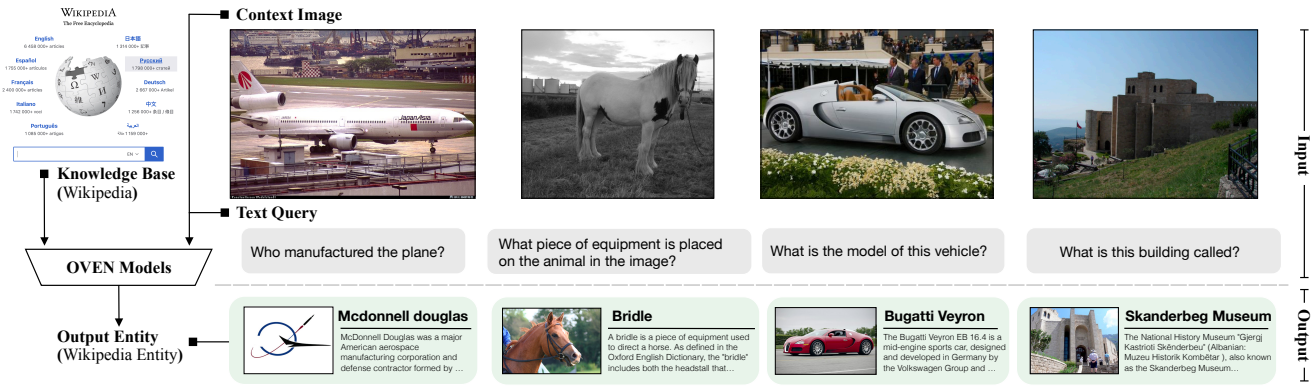


Figure 1: An illustration of the proposed OVEN task. Examples on the right are sampled from the constructed OVEN-Wiki dataset. OVEN aims at recognizing entities *physically presented* in the image or can be *directly inferred* from the image.

Abstract

Large-scale multi-modal pre-training models such as CLIP [30] and PaLI [8] exhibit strong generalization on various visual domains and tasks. However, existing image classification benchmarks often evaluate recognition on a specific domain (e.g., *outdoor images) or a specific task* (e.g., *classifying plant species), which falls short of evaluating whether pre-trained foundational models are* universal *visual recognizers. To address this, we formally present the task of Open-domain Visual Entity recognitioN (*OVEN*), where a model need to link an image onto a Wikipedia entity with respect to a text query. We construct* OVEN-*Wiki* ‡ *by repurposing 14 existing datasets with all labels grounded onto one single label space: Wikipedia entities.* OVEN-*Wiki challenges models to select among six million possible Wikipedia entities, making it a general visual recognition benchmark with the largest number of labels. Our study on state-of-the-art pre-trained models reveals large headroom in generalizing to the massive-scale label space. We show that a PaLI-based auto-regressive visual recognition model performs surprisingly well, even on Wikipedia entities that have never been seen during fine-tuning. We also find existing pre-trained models yield different strengths: while PaLI-based models obtain higher overall performance, CLIP-based models are better at recognizing tail entities.*

1. Introduction

Pre-trained large language models [3, 11], *inter alia*, have shown strong transferable text processing and generation skills in tackling a wide variety of natural language tasks [38, 44, 48] across languages and task formats, while requiring very few manually labeled per-task examples. At the same time, while there has been equally impressive progress in multi-modal pre-training [8, 30], it remains unclear whether similarly universal visual skills, *i.e.*, recognizing millions of coarse-grained and fine-grained visual concepts, have emerged. *Are pre-trained multi-modal models capable of recognizing open-domain visual concepts?*

Answering this question requires a visual recognition dataset with broad coverage of visual domains and tasks, under a universally defined semantic space. Existing recognition benchmarks such as ImageNet [34, 36], Stanford Cars [21], or SUN database [52] represent a large number of visual concepts, but make specific assumptions about the

† Work was done when interned at Google.
‡ Our dataset and evaluation toolkit is publicly available at https://open-vision-language.github.io/oven

granularity of the target concepts (e.g. building type such as "castle" in ImageNet but not a specific building in the world such as "Windsor Castle"), or limit attention to concepts of the same type such as car models/years. Visual question answering (VQA) datasets test models' abilities to recognize concepts which can be of more flexible granularities and object types, but in practice existing VQA datasets tend to focus on higher-level categories. We aim to assess models' abilities to recognize visual concepts from a close to universal, unified space of labels that covers nearly all visual concepts known to humankind, and at a flexible level of granularity, specified by a user or a downstream application. Given a short specification of each element in the target visual concepts space (such as a textual description), multimodal pre-trained models can in principle recognize concepts without seeing labeled instances from each of them.

Towards evaluating models on such universal visual recognition abilities, we introduce the task of **O**pen-domain **V**isual **E**ntity recognitio**N** (OVEN), targeting a wide range of entities and entity granularities, including animals, plants, buildings, locations and much more. Particularly, we construct OVEN-Wiki by building on existing image recognition and visual QA datasets and unifying their label spaces/granularities and task formulations. For our unified label space, we use English Wikipedia which covers millions of visual entities of various levels of granularity and also includes a specification of each entity via its Wikipedia page (containing entity name, text description, images, etc.). Wikipedia also evolves as new entities appear or become known in the world, and can be used as a first approximation of a universal visual concept space.

We re-purpose 14 existing image classification, image retrieval, and visual QA datasets, and ground all labels to Wikipedia. In addition to unifying labels, we unify input recognition intent specifications, which is necessary when combining specialized datasets with the goal of evaluating universal recognition. Given an image showing a car and a tree behind it, OVEN makes the recognition intent explicit via a natural language query such as "What is the model of the car?" or "What is the species of the tree?". Therefore, the OVEN task takes as input an image and a text query[1] that expresses visual recognition intent with respect to the image. The goal is to provide an answer by linking to the correct entity (e.g. BUGATTI VEYRON or BACTRIS GASIPAES) out of the millions of possible Wikipedia entities, each coming with descriptions and a relevant set of images from its Wikipedia page (see Figure 1). Importantly, OVEN requires recognition of entities that were UNSEEN in the training data. Models can still take advantage of the text description and/or images on the Wikipedia page of the UNSEEN entities, as well as knowledge acquired through pre-training.

[1]A query can be expressed in different formats; in this paper, we choose to use a question to reflect the intent.

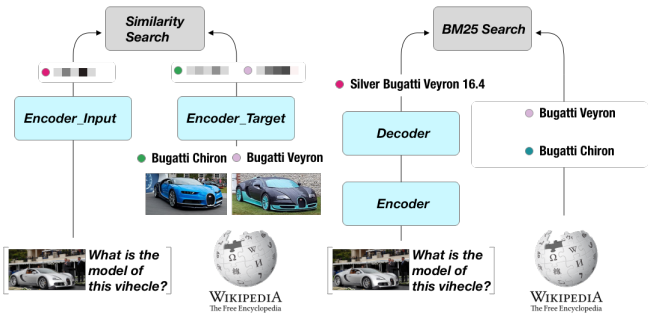
Human annotators were hired to help create OVEN-Wiki for two reasons. First, grounding labels from the component datasets into Wikipedia entities is non-trivial due to language ambiguity. For example, '`Tornado`' can be a weather phenomenon or a type of airplane (PANAVIA TORNADO). To reduce such ambiguity in the grounding, we take multiple steps to refine the labels, including the use of human annotators, a state-of-the-art textual entity linking system [12], and heavy filtering. Second, creating unambiguous textual query intents is also challenging. In many cases, a text query can lead to multiple plausible answers (e.g. of various granularities), and a human often needs to make revisions to make sure no other objects could be correct answers. For our training and development/test sets we rely on semi-automatic processing, but additionally introduce a gold evaluation set, for which annotators thoroughly corrected entity linking errors and rewrote ambiguous input query intents.

Based on OVEN-Wiki, we examine two representative multi-modal pre-trained models, PaLI [8] and CLIP [30], to establish an empirical understanding of the state-of-the-art in universal entity recognition. Particularly, these two models are used for creating an auto-regressive visual entity recognition model (similar to [12]) and a visual entity retrieval model, respectively. Our study suggests that there is a large room for improvement in generalizing to the massive label space. We show that the PaLI-based auto-regressive visual recognition model performs surprisingly well, even on Wikipedia entities that have never been seen during fine-tuning. Digging deeper, we discover that CLIP variants and PaLI-based models make very different kinds of errors. Particularly, PaLI dominates in recognizing popular Wikipedia entities, whereas CLIP models can win consistently on recognizing tail entities.

2. Open Domain Visual Entity Recognition

To drive progress in universal entity recognition, we propose the task of Open-domain Visual Entity recognitioN (OVEN). There are two desiderata that we would like to meet for the OVEN task. First, there should exist a universal label space. In OVEN, we make use of a multi-modal knowledge base, such as Wikipedia, to serve as the universal label space, covering millions of entities. Second, the answer label for each OVEN input should be unambiguous. This is particularly challenging when the label space is very large and multi-granular. To accomplish this, OVEN makes use of input text queries to define the recognition intent (*e.g.*, identifying car types or car models), allowing visual concepts from different granularities to be unambiguously specified.

Task Definition The input to an OVEN model is an image-text pair $x = (x^p, x^t)$, with the text query x^t expressing intent with respect to the corresponding image x^p. Given a unified label space $\mathcal{E}$ which defines the set of all possible



(a) Dual Encoder (b) Encoder Decoder

Figure 2: **Illustration on two typical OVEN Models.**

entities, the knowledge base $\mathcal{K} = \{(e, p(e), t(e)) \mid e \in \mathcal{E}\}$ is a set of triples, each containing an entity e, its corresponding text description $t(e)$ (*i.e.*, name of the entity, description, etc.) and a (possibly empty) set of relevant images $p(e)$. For instance, an entity e = `Q7395937` would have a corresponding textual description $t(e)$ = '`Name: Sabatia campestris; Description:...`'[2] and a set $p(e)$ containing one or more images from the corresponding Wikipedia page[3] of SABATIA CAMPESTRIS. We consider the combination of $t(e)$ and $p(e)$ the *multi-modal knowledge* for the entity e. As OVEN is a recognition task, we focus on recognizing and linking entities that are *physically* present in the image.[4]

The goal of learning for OVEN is to optimize a function f_Θ that predicts the entity e from a given test example $x = (x^p, x^t)$ and the associated knowledge base of triples $\mathcal{K}$. There are different ways to utilize the information available in $\mathcal{K}$, and models may choose to use only a subset of this information. Figure 2 presents two typical ways of modeling OVEN. For encoder-decoder models [8, 49], the most straight-forward utilization is to memorize the entities of the database $\mathcal{K}$ into model parameters Θ via pre-training and fine-tuning, and then *generate* entity names directly during inference. Given that the generated name might not appear in the database, a BM25 search [35] is used to map the prediction to the entity with the closet name in the available database For dual-encoder models [7, 14, 19, 30], an alternative is to explicitly compare a given test example x to representations of entities $e \in \mathcal{E}$, making the prediction an *entity retrieval* problem. We refer to Section 4 for concrete examples of how to implement OVEN models.

Data Split and Evaluation Due to OVEN's goal of evaluating pre-trained multi-modal models, we only provide a partial set of visual concepts (*i.e.*, SEEN categories) for model training or fine-tuning. For evaluation, an OVEN model is tested on generalization to entities not present in the fine-tuning data (thus UNSEEN), without forgetting the SEEN concepts. The models need to either acquire information from the knowledge base, or make a prediction using knowledge obtained during pretraining. We evaluate OVEN with a metric aiming to balance performance between SEEN and UNSEEN entities using a harmonic mean, as shown below:

$$\text{HM}(\text{Acc}_{\text{SEEN}}, \text{Acc}_{\text{UNSEEN}}) = 2 \,/\, \left(\frac{1}{\text{Acc}_{\text{SEEN}}} + \frac{1}{\text{Acc}_{\text{UNSEEN}}}\right) \quad (1)$$

Harmonic mean equally weighs the importance of the SEEN and UNSEEN subsets, and penalizes models with a short barrel. Further details are provided in §3.

OVEN versus recognition benchmarks Given that an OVEN model need to generalize to UNSEEN entities, it is required to predict over all KB entities, which can exceed 6 million in our experiments (*e.g.*, the size of English Wikipedia). This is orders of magnitude larger than existing benchmarks. Second, the large label space has made the generalization to UNSEEN entities the most critical criterion for a successful OVEN model, which also allows future open-domain evaluation[5]. Third, OVEN requires models to do multi-modal reasoning, *i.e.*, comprehending the text query within its visual context, to predict the answer entity.

OVEN versus Visual QA tasks OVEN can be considered as a VQA task because its input format is the same as that of standard VQA models (*e.g.*, text query + image). However, OVEN is specialized and focuses solely on recognition, with the text input serving mainly for intent disambiguation. Moreover, OVEN models are required to generate the name of an entity that exists in a given KB (like models for text entity linking tasks), while VQA models output free-form answers (such as `yes/no` for verification questions and numbers for counting questions).

From OVEN to Knowledge-Intensive VQA Although this paper aims to evaluate pre-trained multi-modal models on universal visual entity recognition, we highlight that models that excel at OVEN can serve as foundational components for systems that can answer knowledge-intensive questions. For example, given an image and a question "When was the church built?", one could apply an OVEN model to link the image to a concrete church's Wikipedia page and then extract the answer from that document. A follow-up work has conducted a thorough study on the value of Wikipedia grounding for answering knowledge-intensive visual questions [9].

3. The OVEN-Wiki dataset

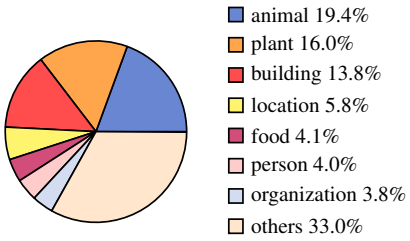
Based on the task formulation of OVEN, we create the OVEN-Wiki dataset by combining 14 existing datasets,

[2] In this paper, we only consider using the name of the entity as its textual representation, despite the fact that more textual descriptions are available.

[3] https://en.wikipedia.org/wiki/File:Sabatia_campestris_Arkansas.jpg

[4] Extending this framework to entities that are not physically present in the image (e.g. the inventor of the airplane) is also valid and useful. See a follow-up works [9] for more details.

[5] One can collect and label a new set of entities from Wikipedia, to serve as a new evaluation data for OVEN models



	Train Set	Val Set	Test Set	Human Set
# unique queries	19,129	3,124	18,341	17,669
# SEEN entities	7,943	1,942	10,137	2,487
# SEEN examples	4,958,569	63,366	366,350	14,016
# UNSEEN entities	0	2,004	10,156	2,174
# UNSEEN examples	0	66,124	362,909	10,851
# Total examples	4,958,569	129,490	729,259	24,867

	Wiki_{EN}
# entities	6,063,945
# images	2,032,340
# title	6,063,945
AvgLen(title)	2.93

Figure 3: Dataset Statistics of the OVEN-Wiki. **Left:** Distribution of super-categories of entities that have positive examples (See Appendix for more details). **Mid:** Statistics of different splits of the OVEN-Wiki. **Right:** Properties of the Wikipedia dump-2022/10/01.

grounding their labels to Wikipedia, resolving label ambiguities, and providing unambiguous textual query intents for all examples. The 14 datasets were originally created for image recognition/retrieval, and visual question answering. Below is the complete list:

- **Image Recognition Datasets**: ImageNet21k-P [34, 36], iNaturalist2017 [45], Cars196 [21], SUN397 [52], Food101 [2], Sports100 [16], Aircraft [26], Oxford Flower [29], Google Landmarks v2 [50].

- **Visual QA Datasets**: VQA v2 [17], Visual7W [55], Visual Genome [22], OK-VQA [28], Text-VQA [42].

These datasets belong to two groups: image recognition (or retrieval) which provides *diverse visual entities*, defined as the **Entity Split** (ES); and VQA which provides *visually-situated natural language queries*, defined as the **Query split** (QS). For examples that originate from VQA datasets, we employ human annotators to write templated rules and filter out questions that do not lead to visual entity answers that are present in the image. For examples from recognition datasets, we first extract the super-category of their label (using the Wikipedia database), and then apply a templated query generation engine to generate a query with unambiguous intent that leads to the label (details in the Appendix).

Label Disambiguation and Human Annotation Grounding the labels of 14 datasets to Wikipedia entities is challenging, and we perform the following steps to accomplish this. We first apply a state-of-the-art textual entity linking system [12] to recognize text labels and map them into Wikipedia. Human annotators are used to write rules to detect bad linking results or unlinkable labels (e.g. numbers), and correct entity linking errors. The union of original dataset labels were linked to 20,549 unique Wikipedia entities, each with a number of examples for the purpose of training and evaluation. Meanwhile, we construct the candidate label space using the English Wikipedia snapshot from *Oct. 1 2022*, by removing all disambiguation, redirect, and media file pages. As shown in Figure 1 (right), this left us with 6,063,945 Wikipedia entities in total. Note that we only consider using the first Infobox images [51] from each page to serve as the visual support for each Wikipedia entity; these are available for 2,032,340 entities.

We further perform human annotation to create a high-quality evaluation dataset. Specifically, we hired over 30 dedicated annotators to validate the entity links in <image, query, answer> triplets sampled from the test split. They were asked to re-annotate the triplets with access to the visual context, ensuring that the query leads to the correct Wikipedia entity answer. Through this process, we collected 24,867 natural language queries, equally distributed over triplets originally sampled from the Entity and Query splits (*i.e.*, test splits). We asked the annotators to rewrite the queries so that no other object in the image could be a valid answer. As a result, the percentage of unique queries in the total examples (17,669 out of 24,867) as shown in Table 3 (mid) is significantly higher in the human set than in the other sets. This brings higher query generalization challenges for the human eval set. We report results using the same evaluation metrics on the human data, with respect to SEEN and UNSEEN entities. Figure 1 provides a glance at the human annotated data.

Dataset Statistics Figure 3 (left) presents the general distribution of the super-categories for our final collection of Wikipedia entities that have positive examples. Figure 3 (right) shows detailed statistics for queries and entities for each of the fine-tuning (train), validation, test, and human splits. Note that the models do not know which entities are present in the val/test/human set, and must scan through the whole KB to make predictions. The # of SEEN/UNSEEN examples indicates the # of examples of which the positive entity labels are in the SEEN/UNSEEN split.

Evaluation Details As aforementioned, we evaluate models by asking them to predict one out of over 6 million English Wikipedia entries. While our data does not cover all 6 million labels as positive examples, models still need to consider all possible outputs due to the presence of UNSEEN entities. We measure the models' performance using both the Entity Split (ES) and Query Split (QS). Specifically, we first compute the harmonic mean of accuracy over examples from the SEEN and UNSEEN classes, as $\text{ACC}_{\text{ES}} = \text{HM}(\text{ACC}_{\text{ESSEEN}}, \text{ACC}_{\text{ESUNSEEN}})$ and $\text{ACC}_{\text{QS}} = \text{HM}(\text{ACC}_{\text{QSSEEN}}, \text{ACC}_{\text{QSUNSEEN}})$ as the Equation 1. Then we further calculate the harmonic mean between splits $\text{HM}(\text{ACC}_{\text{ES}}, \text{ACC}_{\text{QS}})$ to reward models that do well on both

splits. We use the validation data, which contains examples from subsets of both SEEN and UNSEEN entities, for model selection, and we measure performance on the test split and the human evaluation set.

4. Fine-tuning Pre-trained Models for OVEN

We evaluate two prominent pre-trained multi-modal models: CLIP [30], a widely-used dual encoder model for image and text, and PaLI [8], a state-of-the-art pre-trained encoder-decoder model. Figure 2 has illustrated high-levelly on how encoder-decoder and dual encoder models can model the task of OVEN. In the following, we present more details about how these two models can be fine-tuned for OVEN.

4.1. Dual encoders: CLIP and its variants for OVEN

One can naturally apply CLIP on OVEN by treating it as an image-to-text retrieval task. For an input image x^p, the image encoder is used to form an image embedding. Then the predicted entity could be retrieved by finding the entity that has the maximum dot product value between the entity text embeddings and entity image embeddings among the entire entity database. However, this naive implementation ignores the input intent x^t and the entity images $p(e)$.

In the following, we present two variants of CLIPs: CLIP Fusion and CLIP2CLIP. The goal of these two variants is to use all of the information provided in the OVEN task. Both variants learn a function f_Θ that maximizes the score of the target entity for the given input image-query pair, using multimodal knowledge from the knowledge base. Given a test example $x = (x^p, x^t)$ and the knowledge base of triples $\mathcal{K}$, the function is used to make a prediction,

$$e' = \arg\max_{e \in \mathcal{E}} f_\Theta(x^p, x^t, p(e), t(e)) \quad (2)$$

CLIP Fusion adopts the pre-trained CLIP model as the featurizer to develop this system, via adding a 2-layer Multi-Modal Transformer on top of the CLIP image and text features as a mixed-modality encoder. The left encoder (for an input image-query pair) and the right encoder (for multi-modal knowledge information) use the same architecture, but do not share parameters. We fine-tune all of their parameters on the OVEN-Wiki, which includes both the pre-trained CLIP weights and randomly initialized Transformer weights.

CLIP2CLIP relies more heavily on the pre-trained CLIP model and introduces only a minimal set of new parameters (*i.e.*, four) to re-weigh and combine CLIP similarity scores. Particularly, it computes the cosine similarity between $<x^p, t(e)>$, $<x^t, p(e)>$, $<x^p, p(e)>$, and $<x^t, t(e)>$, using the image and text encoders of CLIP, respectively. Then it aggregates these similarities by multiplying them with a learnable vector that reflects importance weights.

Scaling to 6 million candidates. It is expensive to perform dot product scoring with respect to 6 million webpages on-the-fly. Fortunately, there exist approximate algorithms for maximum inner product search whose running time and storage space scale sub-linearly with the number of documents [33, 40, 41]. In all our experiments, we use ScaNN [18] as our library for entity retrieval.

4.2. Encoder-Decoder: PaLI for OVEN

PaLI [8] is a sequence-to-sequence model pre-trained on web text, image-text pairs (*i.e.*, WebLI) and other sources. PaLI can accept both an image and text as input and generates text as output. In order to map the PaLI predictions to the knowledge base, we run a BM25 [35] model to retrieve the most similar Wikipedia entity name for every generated text output. We found that this can slightly but consistently improve the entity recognition results. Note that we directly fine-tune PaLI on the OVEN training data, which does not cover all entities and questions appearing in our Dev and Test splits. However, we found that PaLI is still able to handle entities that are unseen during fine-tuning due to the knowledge acquired during pre-training. To make the comparison with CLIP more comprehensive, we report results on both PaLI-3B and PaLI-17B. The former PaLI variant is at the same magnitude (in its number of parameters) as the largest CLIP model, and the latter PaLI variant is one magnitude larger, and much stronger based on other evaluation [8].

5. Experiments

We first describe the essential experimental setups in §5.1, and then present the main benchmark results in §5.2.

5.1. Experimental Setups

Pre-trained Model Details. For all the CLIP variants, we employ the largest CLIP checkpoints, *i.e.*, `ViT-L14`, which leverages Vision Transformer [13, 46] as its visual backbone. For the PaLI model [8], we make use of the 3B and 17B parameter pre-trained models provided by the original authors, for fine-tuning on OVEN.

Data Processing Details. We process all images in our dataset by resizing them to 224×224, linearize them into a sequence of 14×14 patches, and apply the normalization technique consistent with each model's pretraining to pre-process the images. For natural language text, we perform tokenization based on the adopted pre-trained model's original vocabulary. More details in Appendix.

5.2. Benchmark Results

Main Results Results on the test set are presented in Table 1. There are several interesting (perhaps surprising) observations. First, while CLIP variants (*e.g.*, CLIP Fusion &

[6] The human study is done on a random sampling of 100 examples.

	# Params	Entity Split$_{(\texttt{Test})}$ SEEN	Entity Split$_{(\texttt{Test})}$ UNSEEN	Query Split$_{(\texttt{Test})}$ SEEN	Query Split$_{(\texttt{Test})}$ UNSEEN	Overall$_{(\texttt{Test})}$ HM	Human Eval SEEN	Human Eval UNSEEN	Human Eval HM
Dual Encoders:									
● CLIP$_{\texttt{ViTL14}}$	0.42B	5.6	4.9	1.3	2.0	2.4	4.6	6.0	5.2
● CLIP Fusion$_{\texttt{ViTL14}}$	0.88B	33.6	4.8	25.8	1.4	4.1	18.0	2.9	5.0
● CLIP2CLIP$_{\texttt{ViTL14}}$	0.86B	12.6	10.5	3.8	3.2	5.3	14.0	11.1	12.4
Encoder Decoder:									
◆ PaLI-3B	3B	19.1	6.0	27.4	12.0	11.8	30.5	15.8	20.8
◆ PaLI-17B	17B	28.3	11.2	36.2	21.7	20.2	40.3	26.0	31.6
Human+Search [6]	-	-	-	-	-	-	76.1	79.3	77.7

Table 1: Results of methods on the OVEN-Wiki **test** set and **human evaluation** set. Human+Search represents human performances with information retrieval tools such as search engines and others, on a random subset of OVEN-Wiki$_{\texttt{Human_Eval}}$.

CLIP2CLIP) are utilizing more information from Wikipedia (*i.e.*, entity names and entity images), they are weaker than the auto-regressive PaLI-3B and PaLI-17B model, across most evaluation data splits. This suggests that high-capacity generative multi-modal pre-trained models are capable of recognizing visual entities. Second, this performance gap is more apparent on the query split than the entity split, potentially due to the VQ2A pre-training [6] and the underlying powerful language models [31] in the PaLI model.

Comparing all CLIP-based models, we observe that CLIP Fusion and CLIP2CLIP, which uses all Wikipedia information are generally performing better than the vanilla CLIP model, showcasing the benefits of multimodal information from Wikipedia. Meanwhile, we also observe that CLIP Fusion, where two new layers have been added on top of pretrained CLIP, shows very strong results on SEEN entities for both the Entity and the Query splits, but weak results on UNSEEN entities, thus leading to lower overall performance. The CLIP2CLIP model, on the other hand, is capable of retaining the cross-entity generalization performance while improving its prediction accuracy on SEEN entities.

Comparing the PaLI models, we observe a drastic improvement as the number of parameters in the models increased. Particularly, PaLI-17B has a double-digit performance gain in the overall performances, against the PaLI-3B model. This suggests that scaling the capacity of the model is one of the most important factors, and should be considered as a top priority in future multi-modal dual encoder research.

Results on Human Set and Human Performance. Table 1 shows that the results on the human set are generally aligned with observations on the test set. We conduct a study to estimate the human performance on OVEN-Wiki, via requesting 3 dedicated human annotators to answer 100 examples (sampled from human evaluation set, answers are non-overlapping). We allow the annotators to use search engines (*e.g.*, Google Image Search, Wikipedia Search, etc.)[7], as long as the annotators can provide a valid Wikipedia entity name as the answer. As a result of this study, human achieves 77.7% harmonic mean accuracy, which is significantly higher than the best comparison systems in Table 1.

[7]Even with search engines, each annotator has used 254 seconds to complete one example.

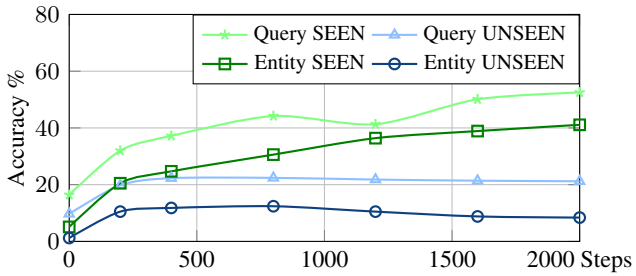


Figure 4: **Fine-tuning PaLI-17B for large # of steps** increases the SEEN accuracy but hurts the UNSEEN accuracy (Results reported on Val split).

6. Analysis

In this section, we perform empirical studies to analyze the pre-trained CLIP2CLIP and PaLI models, and conduct a detailed analysis of these two models' common errors.

Does fine-tuning always help generalization? Figure 4 presents the validation scores of the PaLI-17B model (results of CLIP2CLIP in Appendix), during fine-tuning on OVEN-Wiki's training split. It shows that a longer training schedule does not lead to better generalization performance, particularly when evaluated on the UNSEEN entities. Because of this, we employ the early stopping strategy for model selection, and pick the model with the best harmonic mean combined score on the validation set. However, due to this early stopping strategy, both fine-tuned models are not utilizing 100% of the examples in OVEN's training data because their UNSEEN performance starts to degenerate within one epoch. This has indicated that more advanced fine-tuning strategies that use better regularization techniques to encourage generalization across Wikipedia entities, could be a promising research to explore in the future.

How would the number of entities in KB influence the model's prediction? Figure 5 presents the accuracy of

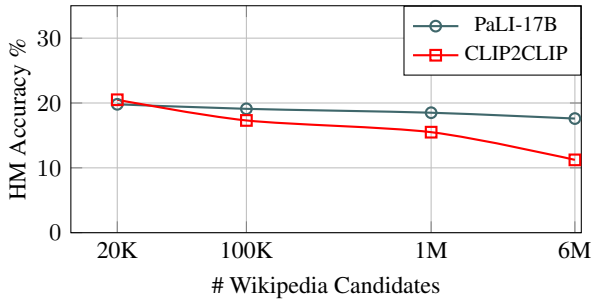


Figure 5: **Impact of # Wikipedia Candidates on PaLI and CLIP2CLIP** (Results reported on the Entity Val split). Increasing the size of Wikipedia makes the tasks difficult.

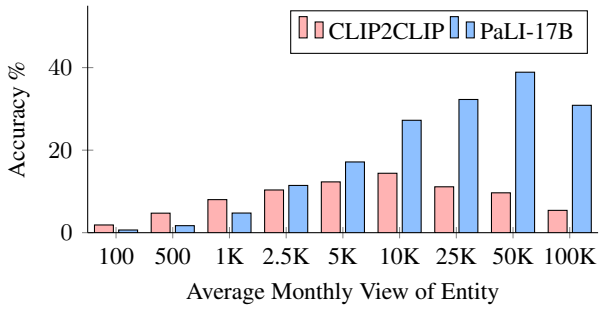


Figure 6: **Comparison on Head vs. Tail Entities** (results on Val split). PaLI wins over CLIP2CLIP on popular (*i.e.*, high monthly page view) Wikipedia entities, but loses on rare (*i.e.*, low monthly page view) Wikipedia entities.

CLIP2CLIP, as a function of the # of total candidates to retrieve from. Here, we compute the accuracy by sub-sampling the negative candidates from KB to different sizes. We observe that when the retrieval candidate entities are only the positive entities (with the # of candidates being 20K), the performance of the CLIP2CLIP model is significantly higher than the open-domain setting (with 6M entities in total). Beyond this, as the KB size increases, model accuracy decreases. Concretely, it shows an approximately linear decline along the log-scale x-axis in Figure 5. This indicates that as the KB size increases, the models' accuracy first drops significantly and then follows with a gradual decline. On the other hand, PaLI's performance is generally more steady as the size of KB grows, potentially because its prediction has already matched up entity names inside KB, so narrowing down the set of candidates does not help the BM25 post-processing. One potential future direction is to employ constrained decoding for the PaLI-based model.

How would models perform on head vs. tail entities? We evaluate the visual entity recognition performances of CLIP2CLIP and PaLI, on entities of different popularity. Particularly, Figure 6 presents a histogram according to models' performance on the entity that has different average monthly Wikipedia page views in 2022 [27]. From the comparison, we can see that PaLI is significantly more accurate compared to CLIP2CLIP, on the head entities (that have more than 5K monthly page views). However, we observe that CLIP2CLIP can perform on par or even outperform PaLI on tail-ish entities (that have less than 2.5K monthly views). This suggests that the retrieval-based visual entity recognition model has its own advantages, in recognizing the difficult and tail entities. Meanwhile, this result suggests that potentially a frequency weighted evaluation should be developed to reward models more with strong recognition capability on the tail entities.

Error analysis To better understand the errors models are making, we sampled a random 100 examples on the human evaluation set, and manually categorize and analyze the errors that PaLI and CLIP2CLIP are making (results in Table 2). Particularly, we categorize the errors of the pre-trained models into four categories: (a) erroneous but relevant prediction, on concepts of the same granularity; (b) errors due to predicting very generic concepts; (c) errors due to misunderstanding the intent behind the query. (d) other miscellaneous errors. Note that errors type (d) are mostly mistakes that are unrelated and not easily interpretable. Figure 7 has provided some concrete examples of the above types of mistakes made by CLIP2CLIP and PaLI. Interestingly, it shows that the two models, *i.e.*, CLIP2CLIP and PaLI, are making very different types of errors in their predictions. Particularly, CLIP based model is good at capturing the right granularity of the entity, but often fails to understand the true intent of the text query. For instance, Figure 7 (c) shows that CLIP2CLIP ignores the text query and starts to predict the name of the barrel racer. In contrast, PaLI is good at following the text query, but can usually predict generic concepts when it does not know the answer confidently.

	PaLI-17B	CLIP2CLIP
Correct	29%	15%
In-correct	71%	85%
→ (a) Wrong But Relevant	23%	27%
→ (b) Too Generic	15%	1%
→ (c) Misunderstand Query	7%	37%
→ (d) Miscellaneous	24%	20%

Table 2: Error statistics for difference models. PaLI predicts more generic answers, while most CLIP errors are due to misunderstanding the questions.

7. Related Works

Learning to Recognize UNSEEN Categories A large body of prior work [23, 25, 47] has studied the generalization to novel categories at test time. Zero-shot learning (ZSL) is one of such attempts that tackles learning new categories with zero images for training. Particularly, ZSL methods rely generating UNSEEN image classifiers from semantic representations, in the format of manually labeled attributes [23], unsupervised learned word vectors [5], or pre-trained sentence embeddings [20, 30]. Few-shot learning (FSL) [47] proposes a more realistic setup, where learners see limited number of visual exemplars during the model deployment.

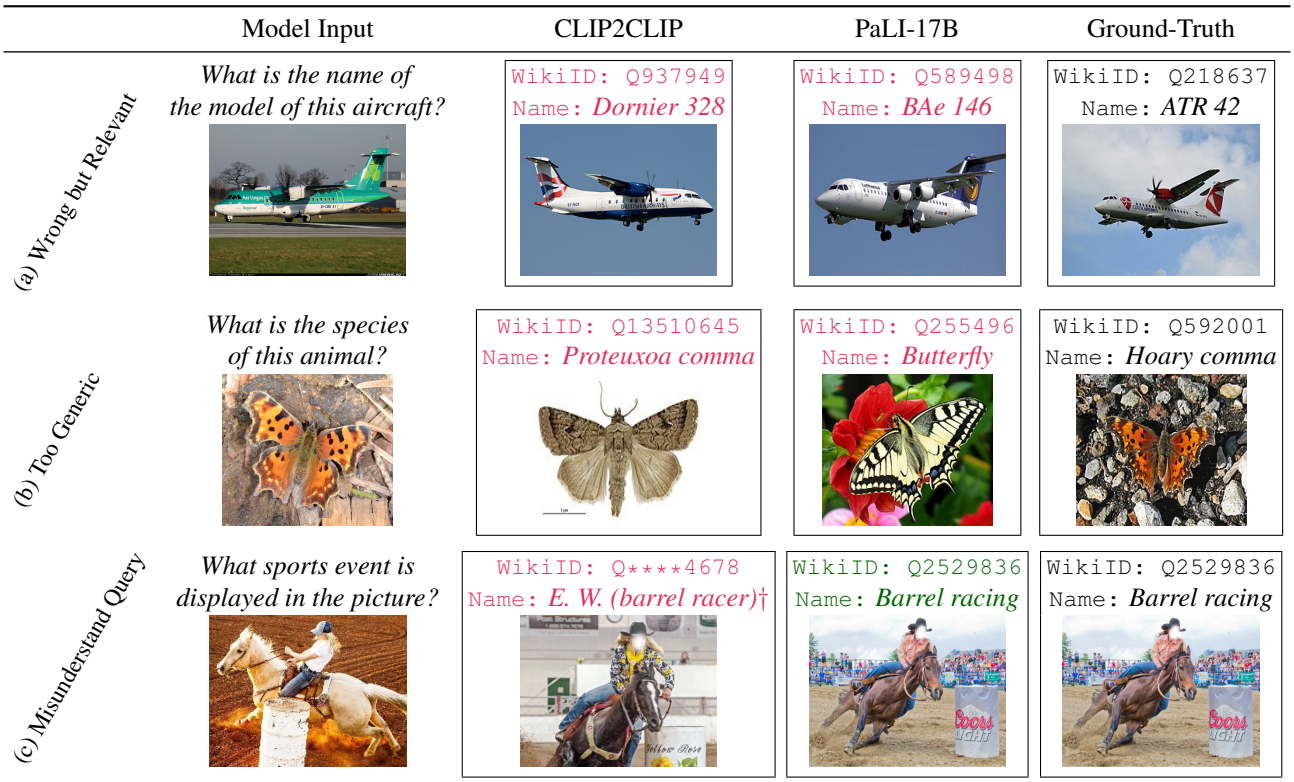


Figure 7: **Visualization of mistakes made by the CLIP2CLIP and PaLI-17B Model.** We visualize the Wikipedia infobox images for each of model's predictions, to provide more context about the visual similarity between the prediction/ground-truth and the input image. Correct predictions are marked as green, whereas incorrect predictions are marked as red. (†: Since no infobox image is available for this Wikipedia entity, a face-anonymized Web image of the entity is visualized for reference.)

With this goal, FSL methods extract the inductive bias of learning the SEEN classes [15, 32, 37, 43, 54], such that the model can leverage it in learning the UNSEEN classes, to avoid severe over-fitting. Comparing to them, OVEN exposes different challenges as we ask the model to make the best use of Web knowledge (*i.e.*, Wikipedia pages with images), which contains textual semantic information and visual appearance of the open-world entities.

Vision & Language + Knowledge There are many efforts combining knowledge into vision-language tasks, such as VQA [4, 10, 28, 39] and entity-focused image captioning [1, 24]. Among them, knowledge-based VQA is most related to OVEN, but also differs in many aspects. Specifically, [4] presents a text QA dataset that requires understanding multi-modal knowledge in a KB. [39] propose to perform knowledge-based VQA tasks, centered around relational questions over public Figures. Later, [28] propose to answer questions where the answer is outside of the image context, to assess model's understanding of real-world. More recently, [10] studies the zero-shot VQA setting where some answers (out of the total 500 frequent answers) are unseen. Comparing to them, OVEN steps back to the more fundamental problem of linking visual content and KB entity, but at a larger scale and broader coverage.

8. Discussion

We have introduced OVEN, a task that aims to unambiguously link visual content to the corresponding entities in a web-scale knowledge base (*i.e.*, Wikipedia), covering a total of more than 6 millions of entities. To facilitate the evaluation of OVEN, we created the OVEN-Wiki dataset, via combining and re-annotating 14 existing visual recognition, retrieval, and visual QA datasets, and linked over 20K labels to the Wikipedia entities. With OVEN-Wiki, we evaluate state-of-the-art multi-modal pre-trained models, *i.e.*, the CLIP [30]-based entity retrieval models and the PaLI [8]-based entity generation model, via fine-tuning them for the OVEN task, to examine their capability on recognizing open-domain visual concepts. As a result, PaLI models have presented significantly stronger performances than the CLIP variants, even on unseen visual entities during the fine-tuning. While the CLIP-based entity retrieval model is overall weaker, it shows advantages in recognizing the tail visual entities. One additional nice property of OVEN-Wiki is its strong extensibility. Through grounding all recognition labels to Wikipedia entities, we as a community can keep growing the member recognition datasets of OVEN-Wiki, by adding positive instances to Wikipedia entities that do not have examples by far. Moreover, successful OVEN models

can generalize to recognize emerging entities (*e.g.*, `iPhone 14 Pro`), as long as the corresponding Wikipedia page is created. In summary, we hope OVEN will drive future research on knowledge-infused multimodal representation learning.

Ethics Statement

As our dataset, *i.e.*, OVEN-Wiki, is composed of existing image recognition, image retrieval, and visual question answering datasets, we have introduced minimum risk of exposing additional social bias in our data. However, OVEN-Wiki is still at the risk of inheriting existing dataset biases. As a result, we employed existing data curation strategies [53] to reduce such potential risks. Besides such risk, OVEN-Wiki also opens up new possibilities that can alleviate ethical concerns in AI systems. Specifically, OVEN-Wiki is a dataset that targets advancing research for establishing stronger grounding between the visual content and knowledge base, which can potentially contribute to building more attributed visual systems, such as a visual question answering model that produces answers based on the linked Wikipedia page, with improved interpretability and controllability.

Acknowledgement

We thank Boqing Gong, Soravit Changpinyo for reviewing on an early version of this paper in depth, with valuable comments and suggestions. We thank Xi Chen for providing different variants of PaLI pre-trained checkpoints. We also thank Radu Soricut, Anelia Angelova, Fei Sha, Alan Ritter, Chao-Yuan Wu, Jiacheng Chen for discussions and feedback on the project.

References

[1] Ali Furkan Biten, Lluis Gomez, Marçal Rusinol, and Dimosthenis Karatzas. Good news, everyone! context driven entity-aware captioning for news images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12466–12475, 2019. 8

[2] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – mining discriminative components with random forests. In *ECCV*, 2014. 4

[3] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. 1

[4] Yingshan Chang, Mridu Narang, Hisami Suzuki, Guihong Cao, Jianfeng Gao, and Yonatan Bisk. Webqa: Multihop and multimodal qa. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16495–16504, 2022. 8

[5] Soravit Changpinyo, Wei-Lun Chao, Boqing Gong, and Fei Sha. Synthesized classifiers for zero-shot learning. In *CVPR*, pages 5327–5336, 2016. 7

[6] Soravit Changpinyo, Doron Kukliansky, Idan Szpektor, Xi Chen, Nan Ding, and Radu Soricut. All you may need for vqa are image captions. *arXiv preprint arXiv:2205.01883*, 2022. 6

[7] Jiacheng Chen, Hexiang Hu, Hao Wu, Yuning Jiang, and Changhu Wang. Learning the best pooling strategy for visual semantic embedding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15789–15798, 2021. 3

[8] Xi Chen, Xiao Wang, Soravit Changpinyo, AJ Piergiovanni, Piotr Padlewski, Daniel Salz, Sebastian Goodman, Adam Grycner, Basil Mustafa, Lucas Beyer, et al. Pali: A jointly-scaled multilingual language-image model. *arXiv preprint arXiv:2209.06794*, 2022. 1, 2, 3, 5, 8

[9] Yang Chen, Hexiang Hu, Yi Luan, Haitian Sun, Soravit Changpinyo, Alan Ritter, and Ming-Wei Chang. Can pre-trained vision and language models answer visual information-seeking questions? *Technical report*, 2023. 3

[10] Zhuo Chen, Jiaoyan Chen, Yuxia Geng, Jeff Z Pan, Zonggang Yuan, and Huajun Chen. Zero-shot visual question answering using knowledge graph. In *International Semantic Web Conference*, pages 146–162. Springer, 2021. 8

[11] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022. 1

[12] Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. Autoregressive entity retrieval. *arXiv preprint arXiv:2010.00904*, 2020. 2, 4

[13] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 5

[14] Fartash Faghri, David J Fleet, Jamie Ryan Kiros, and Sanja Fidler. Vse++: Improved visual-semantic embeddings. In *BMVC*, 2017. 3

[15] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, 2017. 8

[16] Gerry. Sports100: 100 sports image classification. https://www.kaggle.com/datasets/gpiosenka/sports-classification/metadata, 2021. Accessed: 2022-09-26. 4

[17] 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 *CVPR*, 2017. 4

[18] Ruiqi Guo, Philip Sun, Erik Lindgren, Quan Geng, David Simcha, Felix Chern, and Sanjiv Kumar. Accelerating large-scale inference with anisotropic vector quantization. In *International Conference on Machine Learning*, 2020. 5

[19] 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 *International Conference on Machine Learning*, pages 4904–4916. PMLR, 2021. 3

[20] Jihyung Kil and Wei-Lun Chao. Revisiting document representations for large-scale zero-shot learning. In *EMNLP*, 2021. 7

[21] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *3dRR-13*, 2013. 1, 4

[22] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123(1):32–73, 2017. 4

[23] Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. Attribute-based classification for zero-shot visual object categorization. *TPAMI*, 36(3):453–465, 2014. 7

[24] Fuxiao Liu, Yinghan Wang, Tianlu Wang, and Vicente Ordonez. Visual news: Benchmark and challenges in news image captioning. *arXiv preprint arXiv:2010.03743*, 2020. 8

[25] Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X Yu. Large-scale long-tailed recognition in an open world. In *CVPR*, 2019. 7

[26] S. Maji, J. Kannala, E. Rahtu, M. Blaschko, and A. Vedaldi. Fine-grained visual classification of aircraft. Technical report, 2013. 4

[27] Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Hannaneh Hajishirzi, and Daniel Khashabi. When not to trust language models: Investigating effectiveness and limitations of parametric and non-parametric memories. *arXiv preprint arXiv:2212.10511*, 2022. 7

[28] Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. Ok-vqa: A visual question answering benchmark requiring external knowledge. In *Proceedings of the IEEE/cvf conference on computer vision and pattern recognition*, pages 3195–3204, 2019. 4, 8

[29] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*. IEEE, 2008. 4

[30] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 1, 2, 3, 5, 7, 8

[31] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020. 6

[32] Aniruddh Raghu, Maithra Raghu, Samy Bengio, and Oriol Vinyals. Rapid learning or feature reuse? towards understanding the effectiveness of maml. 2020. 8

[33] Parikshit Ram and Alexander G Gray. Maximum inner-product search using cone trees. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 931–939, 2012. 5

[34] Tal Ridnik, Emanuel Ben-Baruch, Asaf Noy, and Lihi Zelnik-Manor. Imagenet-21k pretraining for the masses. In *NeurIPS*, 2021. 1, 4

[35] Stephen Robertson, Hugo Zaragoza, et al. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389, 2009. 3, 5

[36] 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, 4

[37] Andrei A Rusu, Dushyant Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, and Raia Hadsell. Meta-learning with latent embedding optimization. In *ICLR*, 2019. 8

[38] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4938–4947, 2020. 1

[39] Sanket Shah, Anand Mishra, Naganand Yadati, and Partha Pratim Talukdar. Kvqa: Knowledge-aware visual question answering. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 8876–8884, 2019. 8

[40] Fumin Shen, Wei Liu, Shaoting Zhang, Yang Yang, and Heng Tao Shen. Learning binary codes for maximum inner product search. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4148–4156, 2015. 5

[41] Anshumali Shrivastava and Ping Li. Asymmetric lsh (alsh) for sublinear time maximum inner product search (mips). In *Advances in Neural Information Processing Systems*, pages 2321–2329, 2014. 5

[42] Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *CVPR*, 2019. 4

[43] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *NeurIPS*, 2017. 8

[44] Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*, 2022. 1

[45] 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 *CVPR*, 2018. 4

[46] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 5

[47] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. *Advances in neural information processing systems*, 29, 2016. 7

[48] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2018. 1

[49] Zirui Wang, Jiahui Yu, Adams Wei Yu, Zihang Dai, Yulia Tsvetkov, and Yuan Cao. SimVLM: Simple visual language model pretraining with weak supervision. In *ICLR*, 2021. 3

[50] Tobias Weyand, Andre Araujo, Bingyi Cao, and Jack Sim. Google landmarks dataset v2-a large-scale benchmark for

instance-level recognition and retrieval. In *CVPR*, 2020. 4

[51] Inc. Wikipedia Foundation. Help:infobox picture. https://en.wikipedia.org/wiki/Help:Infobox_picture. 4

[52] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *CVPR*, 2010. 1, 4

[53] Kaiyu Yang, Klint Qinami, Li Fei-Fei, Jia Deng, and Olga Russakovsky. Towards fairer datasets: Filtering and balancing the distribution of the people subtree in the imagenet hierarchy. In *Proceedings of the 2020 conference on fairness, accountability, and transparency*, pages 547–558, 2020. 9

[54] Han-Jia Ye, Hexiang Hu, De-Chuan Zhan, and Fei Sha. Few-shot learning via embedding adaptation with set-to-set functions. In *CVPR*, 2020. 8

[55] Yuke Zhu, Oliver Groth, Michael Bernstein, and Li Fei-Fei. Visual7w: Grounded question answering in images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016. 4