Webly Supervised Concept Expansion for General Purpose Vision Models

Amita Kamath* [1] Christopher Clark* [1] Tanmay Gupta* [1]
Eric Kolve[1] Derek Hoiem[2] Aniruddha Kembhavi[1]
[1]PRIOR @ Allen Institute for AI [2]University of Illinois at Urbana-Champaign
https://prior.allenai.org/projects/gpv2

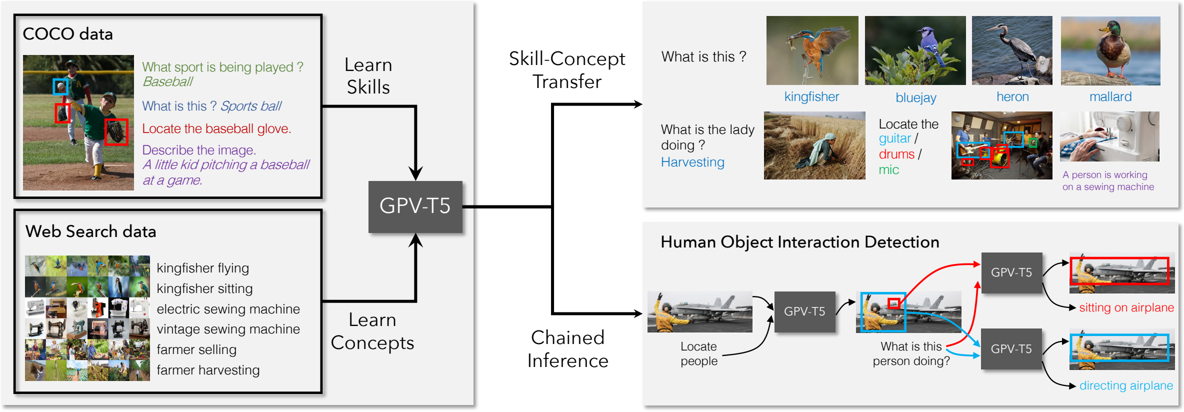


Figure 1. **Learning concepts from the web with GPV-T5.** We propose expanding concept knowledge of general purpose vision systems by learning skills from supervised datasets while learning concepts from web image search data. Existing GPVs are able to effectively transfer webly-supervised concepts across skills such as captioning, classification, and localization. Web data greatly benefits our proposed GPV-T5 architecture by expanding its vocabulary across skills. GPV-T5 supports niche tasks like Human-Object Interaction detection that require multi-step inference without any architectural modifications.

Abstract

*General purpose vision (GPV) systems [25] are models that are designed to solve a wide array of visual tasks without requiring architectural changes. Today, GPVs primarily learn both skills and concepts from large fully supervised datasets. Scaling GPVs to tens of thousands of concepts by acquiring data to learn each concept for every skill quickly becomes prohibitive. This work presents an effective and inexpensive alternative: learn skills from fully supervised datasets, learn concepts from web image search results, and leverage a key characteristic of GPVs – the ability to transfer visual knowledge across skills. We use a dataset of 1M+ images spanning 10k+ visual concepts to demonstrate webly-supervised concept expansion for two existing GPVs (*GPV-1 *[25] and* VL-T5 *[14]) on 3 benchmarks - 5* COCO *based datasets (80 primary concepts), a newly curated series of 5 datasets based on the OpenImages and VisualGenome repositories (∼500 concepts) and the Web-derived dataset (10k+ concepts). We also propose a new architecture,* GPV-T5 *that supports a variety of tasks – from vision tasks like classification and localization to vision+language tasks like QA and captioning to more niche ones like human-object interaction recognition.* GPV-T5 *benefits hugely from web data, outperforms* GPV-1 *and* VL-T5 *across these benchmarks, and does well in a 0-shot setting at action and attribute recognition.*

1. Introduction

While much work in computer vision has focused on building task[1] specific models [28, 35, 66], there has been a recent push towards building more general purpose vision systems (GPVs) [14,25,31]. In contrast to specialized models, GPVs aim to natively support learning a wide variety of

[1] Concepts, skills and tasks are defined as follows: **Concepts** – the union of nouns, verbs and adjectives (*e.g. car, running, red*), **Skills** – operations that we wish to perform on the given inputs (*e.g.* classification, object detection, image captioning), **Tasks** – predefined combinations of a set of skills performed on a set of concepts (*e.g.* ImageNet classification task involves the skill of image classification across 1000 concepts).
* Equal contribution.

tasks, generalize learned skills and concepts to novel skill-concept combinations, and to learn new tasks efficiently.

Today, GPVs are trained and evaluated on strongly supervised datasets such as COCO [49] and VISUALGENOME [40] that expose models to various skill-concept combinations. For instance, learning localization from COCO exposes the models to 80 concepts for that skill. In this paradigm, expanding the model's concept vocabulary to new concepts requires collecting fully-supervised task data for each of those concepts. We wish to build GPVs that can perform a variety of tasks like localization, classification and VQA across more than 10,000 concepts, two orders of magnitude more than the primary concept count of COCO.

Given the large cost of producing high-quality datasets, merely scaling today's manually annotated datasets to support 10,000+ concepts is infeasible. We present an effective and inexpensive alternative for concept expansion: learn skills like localization and VQA from present day vision datasets; learn a massive number of concepts using data from image search engines; and use architectures that can effectively transfer learned concepts across acquired skills. Image search engines provide remarkably good results for millions of queries by leveraging text on the accompanying web pages, visual features extracted from images, and click data obtained by millions of users querying and selecting relevant results each day. They often provide high-quality object and action-centric images, decluttered from distractions, which can be used to learn powerful visual representations for concepts. Importantly, searches scale easily and inexpensively to thousands of queries. We use Bing Search to collect a dataset with 1M+ images, named WEB10K, spanning roughly 10k nouns, 300 verbs, and 150 adjectives along with thousands of noun-verb and noun-adj combinations, costing just over $150.

Although search engine data provides strong supervision only for the task of classification, we demonstrate that current GPVs, GPV-1 and VL-T5, are able to learn concepts from web data and improve on other skills such as captioning. We further build on these models and propose GPV-T5, a powerful general purpose vision system with support for a broader set of modalities (and hence tasks). GPV-T5 can accept as input an image, a task description, and a bounding box (allowing the user to point at an object or region of interest), and can output text for any bounding box or for the entire image. These diverse input and output modalities enable GPV-T5 to support a large spectrum of skills ranging from vision skills like classification and localization, vision-language skills like VQA and captioning to niche ones like classification in context and human-object interaction detection. An important design principle of GPV-T5 is that *all* tasks are based on scoring/ranking/generation using the same text decoder applied to one or more image regions, so that all tasks share the same weights and representations. We also propose a simple re-calibration mechanism to downweight scores of labels that are disproportionally represented in training.

We evaluate these GPVs on three benchmarks: (i) the COCO-SCE and COCO benchmarks [25], designed to test the skill-concept transfer ability and overall skill competency on 80 primary COCO concepts across 5 skills; (ii) a new benchmark, named DCE that is based on the OPENIMAGES and VISUALGENOME datasets for broader concept evaluation for the same 5 skills but now across 492 OPENIMAGES concepts instead of the 80 present in COCO; and (iii) the WEB10K dataset consisting of images from Bing Image Search paired with questions and answers that covers 10,000+ concepts. Our analysis shows that all three GPVs benefit from web data. Furthermore, GPV-T5 outperforms both GPV-1 and VL-T5 across these benchmarks and shows significantly large gains when using web data, particularly for captioning and classification. GPV-T5 also performs well in a 0-shot setting at downstream tasks like action and visual attribute recognition. Lastly, we demonstrate how GPV-T5 can be chained to perform niche tasks like human-object interaction detection, without any task-specific architecture modifications.

In summary, our main contributions include: (a) WEB10K, a new web data source to learn over 10k visual concepts with an accompanying human-verified VQA benchmark; (b) demonstration of concept transfer from WEB10K to other tasks; (c) DCE, a benchmark spanning 5 tasks and approximately 500 concepts to evaluate GPVs; and (d) GPV-T5, an architecture that supports box and text modalities in both input and output, improves skill-concept transfer by sharing the same encoders and decoder for all tasks and using classifier re-calibration, outperforms existing GPVs, and achieves reasonable zero-shot generalization to visual attribute and verb recognition tasks. Our code and datasets will be publicly released.

2. Related Work

General purpose models. Computer vision models have progressively become more general. Specialization first gave way to multitask models which aimed at solving multiple, albeit predefined, tasks with one architecture. A common approach for building such models [27, 52] is to use task-specialized heads with a shared backbone. However, adding a new head for each new task makes scaling to a large number of tasks and reuse of previously learned skills challenging. An alternative approach is to build a *general-purpose* architecture that can be used for many tasks without task-specific components. This approach has become common in natural language processing via text-to-text generative models [5, 55, 64], and recent work in computer vision has striven towards this kind of generality [7,17,37,50].

Examples of general-purpose computer vision models include VL-T5 [14], which adapts T5 [64] to jointly train on vision-language tasks while using a single text-generation head to produces outputs for all tasks, and GPV-1 [25], which combines a similar text-generation head with the ability to return bounding-boxes and relevance scores as output. In this work, we work with both GPV-1 and VL-T5 and extend their concept vocabulary with web data. Our proposed model, GPV-T5 follows VL-T5 in its use of the T5 backbone, builds upon the vision capabilities of GPV-1, and further extends the range of tasks that can be performed by allowing a bounding-box to be used as input and introducing the ability to generate per-image-region text output. Works such as Perceiver [31] and PerceiverIO [30] aim to generalize the architecture beyond images and text to other modalities such as audio, video, and point cloud. However, both architectures remain to be tested for multitask learning and for learning vision-language tasks such as VQA and captioning. Many other V+L models [13, 47, 53, 74, 84] can be fine-tuned on a variety of downstream tasks, but they typically use task-specific heads, while the focus of our work is on general purpose models in a multi-task setting.

Web supervision. Image search engines provide highly relevant results, using a combination of text, image and user features. Researchers have used search data as a form of supervision to build computer vision models. Early works used noisy retrieved results with probabilistic Latent Semantic Analysis [20] and multiple instance learning [78] to build recognition systems. As web results improved, works used this data to build object detectors [11, 15, 46, 54, 71, 83], attribute detectors [21], image taggers [81], large vocabulary categorization models [24, 56, 85] and fine-grained recognition models [39, 57], segmentation models [33, 70, 73], online dataset builders [45], visual reasoning systems [92] and visual knowledge bases with learnt relationships between objects [12]. More recently, massive scale web data in the form of retrieved search results and the accompanying text was employed to build the powerful CLIP family of models [62] that provide powerful visual representations for downstream tasks. While these works have shown that web data can be used to build single task models, we show that one can build GPV's with web data and importantly transfer this knowledge across skills.

Concept transfer across skills. There has been considerable interest in transferring concept knowledge from classification to object detection since classification labels are far cheaper to obtain than detection labels. Hoffman *et al*. [29] cast this problem as a domain adaptation problem, adapting classifiers to detectors. Redmon *et al*. [67] jointly train for the two tasks using fully and weakly supervised losses, enabling them to train a 9,000 class real time detector using Imagenet22k classification data [16]. Uijlings *et al*. use Multiple Instance Learning to pseudo label data and then train a large vocabulary detector. Recent works build open vocabulary detectors [23, 32, 88] by leveraging image caption pairs (or models like CLIP [63] which are built from the same), obtained in large quantities on the web. Even though image-captions are noisy, the resulting detectors improve as the data is scaled up.

The vision+language field has leveraged pre-trained object detectors as feature inputs for tasks like VQA and captioning [2, 3, 90]. This can be considered as transferring visual concept knowledge from object detection to downstream tasks. Another effective approach is pre-training using image-captions [44, 47, 51] like Conceptual Captions [68]. CLIP [63] is a family of powerful models that are pre-trained on a massive 400M image caption paired dataset. The resulting encoders are very effective at vision+language tasks [69]. These methods effectively transfer visual knowledge from caption data to tasks like VQA. Recently Whitehead *et al*. [82] disentangle the encoding of concepts and skills and build a model that can generalize to new skill-concept compositions and new concepts for VQA.

The focus of our work is to build a GPV that can transfer concepts across various skills, particularly from web data to vision and vision-and-language skills, and also provide a new test-only evaluation benchmark for the same.

3. The WEB10K dataset

We present WEB10K, a dataset sourced from web image search data with over 10K concepts. The two primary advantages of search engine data are: (1) Search engines benefit from a large volume of user click data to produce high-quality results for a large vocabulary of concepts including tail concepts not frequently mentioned in annotated computer vision datasets (e.g. "hyacinth"); and (2) The image distribution of search engine results tends to be similar to image classification data with the image centered on the queried object with few distractions, making them ideal for learning visual concept representations. WEB10K contains queries with nouns, adjectives and verbs.

Nouns. We consider single-word and multi-word nouns. Single-word nouns are sourced from a language corpus with a list of 40,000 concrete words [6], each with a concreteness score (defined as the degree to which a word refers to a perceptible entity). From this list, we select nouns with a concreteness score $> 4.0/5$ and any verb or adjective with an alternate word sense as a noun (e.g. "comb") with a score $> 4.5/5$. These thresholds avoid more abstract or non-visual words such as "humor". Multi-word nouns are sourced from CONCEPTUAL CAPTIONS (CC) [68]. We identify candidates using POS tagging and select the most frequent 2,000, and select an additional 282 where the second word of the multiword noun is present in the concreteness dataset (e.g. "street artist", where "artist" is in concrete nouns). In total, we select 10,213 nouns. Sourcing nouns

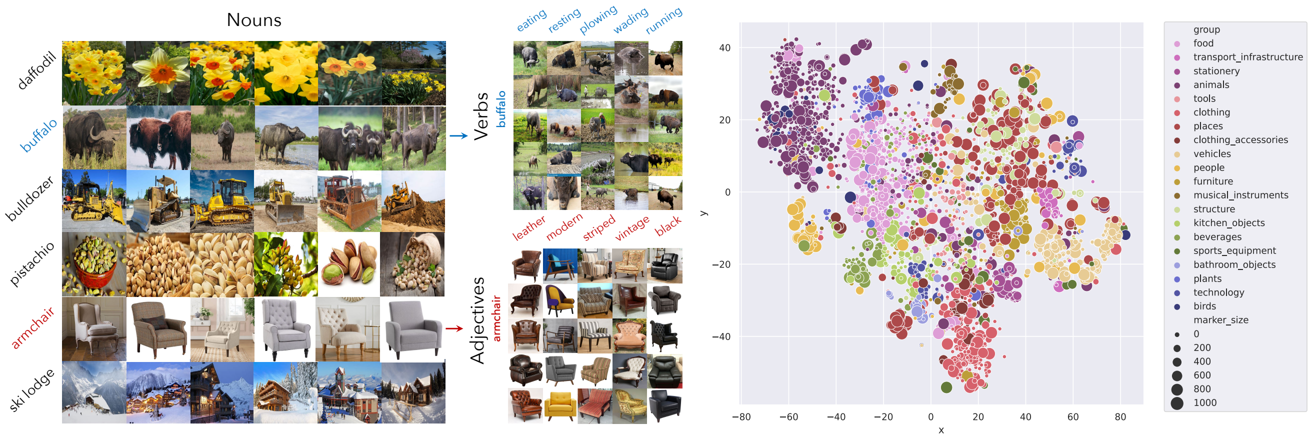


Figure 2. **Concept diversity in WEB10K. Left:** Besides 10k nouns, WEB10K provides dense coverage of feasible adj-noun and verb-noun combinations to enable learning of fine-grained differences in object appearance due to attributes. **Right:** TSNE [76] plot of Phrase-BERT [80] embeddings of nouns with bubble size indicating frequency (capped at 1000) in CC, a common large-scale pretraining dataset. WEB10K nouns cover a wide range of concept groups identified using WordNet and include many which are infrequent/absent in CC.

from the Concreteness corpus enables coverage of concepts not commonly covered in vision datasets: >4,000 nouns in WEB10K are not present in CC, e.g. "wind tunnel".

Verbs. We source verbs from a combination of vision datasets with large verb vocabularies including imSitu [86], HICO [9] and VRD [48]. We remove verbs that are either polysemous (have multiple meanings e.g. "person holding breath" vs. "person holding cup") or aren't associated with an animate agent (e.g. "snowing"). This results in 298 verbs that are unambiguous and visually recognizable.

Adjectives. We source adjectives from several datasets that have a large number of adjectives [10, 19, 40, 41, 43, 59, 60, 68, 79]. We manually filter out ones that are subjective ("beautiful"), non-visual ("loud"), or relative ("big"). This results in 148 adjectives which we group into 16 adjective types (e.g. "color", "texture").

We select noun-adj pairs and noun-verb pairs which appear at least thrice in CC: this removes nonsensical pairs, e.g. "cloudy dog". The total number of queries in WEB10K is 38,072 with roughly 10k nouns, 18k noun-adj and 9k noun-verb combinations. We feed each query into the Bing Search API and retrieve a total of 950,443 image URLs (approx. 25 per query). **Importantly, this cost us $154**, so it is inexpensive to scale further, and such data acquisition is affordable for many other research organizations.

Conversion into QA data. We convert each query-image pair into multiple templated QA pairs where the answer is the noun, adjective or verb from the query. For example "What is the [noun] doing?" and "What [adj type] is this object?"; see supplementary for all question templates. This QA format has two advantages: (1) it removes ambiguity from the task (e.g., "What color is this" tells the model not to return a potentially accurate non-color attribute); and (2) it bridges the domain gap to other tasks posed as questions.

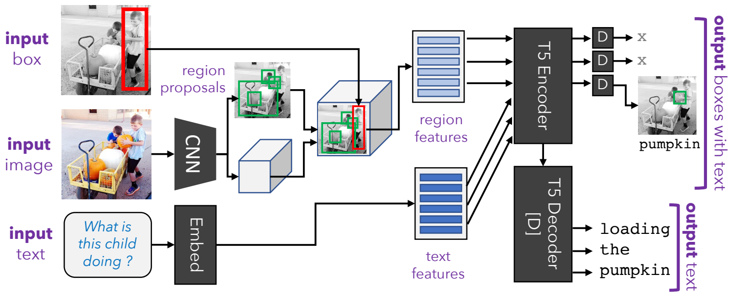
Data Splits. We split image-query pairs into train (874k), val (38k) and test (38k). We sample 5k and 10k pairs from the val and test sets and ask 3 crowdworkers to verify that the query is present in the image. We only retain unanimously verified examples (71%) resulting in a val set of 4k images (9k QAs) and a test set of 8k images (19k QAs). The train set has 1.5M QAs with no manual verification.

4. GPV-T5

In this section we present our GPV model, GPV-T5. Following VL-T5, GPV-T5 combines an object detector with the T5 pre-trained language model. GPV-T5 supports additional input and output modalities (and thus tasks) beyond present day GPVs (GPV-1 and VL-T5). It uses the stronger VinVL [90] object detector, uses a shared language decoder (for all tasks including localization) and employs a classification re-calibration approach that together improve generalization to unseen concepts at test time.

Model design. GPV-T5 takes an image, text, and bounding box as input. As output, it can produce text for an individual bounding box (including the input one or ones produced by the visual model) and for the entire image (see Figure 3).

First, the input text is tokenized and embedded using T5-Base to get a sequence of text feature vectors. Then an object detection model is used to identify regions in the image and extract bounding boxes and features for those regions (we do not use the class labels identified by the detector) via RoI pooling. We additionally use the object detector to extract features for the input bounding box, and a learned embedding is added to those features to distinguish them from the other visual features. These sets of visual features are then converted to embeddings of the same dimensionality as the text embedding using a linear layer. We primar-



	INPUTS			OUTPUTS	
Skill	Image	Text/Prompt	Bbox	Text	Bbox+Scores
VQA	Full	Question	-	Answer	Attended regions
Cap	Full	Describe the image Caption this image	-	Caption	Attended regions
Loc	Full	Find {OBJECT} Locate instances of [OBJECT]	-	-	Localized [OBJECT] instances
Cls	Cropped	What is this thing? What object is this?	-	Category	-
CiC	Full	What is this thing? What object is this?	Region to classify	Category	-

Figure 3. **Left**: The GPV-T5 model architecture. **Right**: I/O for the 5 skills in COCO and DCE evaluation.

ily use the VinVL [90] object detector for our experiments. However the GPV-T5 architecture allows us to easily swap in other detectors, and we use features from the DETR [7] object detector for some of our experiments in Sec. 6.

The resulting visual and text vectors are concatenated as a sequence and used as an input to the T5-Encoder to build joint contextualized embeddings. To generate text for the entire image we use the T5-Decoder with this contextualized embedding sequence as input, and to generate text for individual boxes we run the same T5-Decoder while using the contextualized embeddings that corresponds to just that box as input. The usage of a common decoder for image-based outputs and region-based outputs enables transfer of learned concepts between skills that require processing the entire image and skills that rely primarily on the representation of a single region.

Using GPV-T5. GPV-T5's design gives us flexibility to handle a variety of vision and vision+language tasks without needing task-specific heads. For tasks that do not have text input (e.g. classification) we follow [25] by building appropriate text prompts for that task (e.g., "What is this object?" for classification) and selecting one at random to use as the input text. For tasks that do not have an input bounding box, we use a box around the entire image.

Decoded text from the image is used to answer questions and generate captions. For classification or limited-choice responses, answers are scored based on log-probability of generating each option, and the highest scoring answer is chosen. To localize objects, we propose Language-Based Localization (LBL) where the score of a box is computed by first computing the log-probabilities of generating an object class or "other" from that box, and then applying a linear classifier to those scores to a scalar relevance score. For example, "Localize dog" is performed by computing the log-probability of "dog" and "other" for each region.

Importantly, the same text decoder is used to generate image and region text, so that *classification, question answering, captioning, localization, and all other tasks use the same encoders, decoder, and weights*. Our experiments show that this facilitates skill-concept transfer.

Even complex tasks like human-object interaction (HOI) can be performed by chaining inference steps (Fig. 1). HOI [8, 9] requires localizing a person, an object and categorizing their interaction. GPV-T5 performs this by first returning detections for "Locate person", then providing each person box as input with the prompt "What is this person doing?" The log-probs of generating object-interaction phrases, such as "direct the airplane" for other boxes are used to identify the most likely interaction.

Classification re-calibration. We observe that a common issue in classification is that the model becomes biased towards classes that are common in the training data. For example, we find that if the model is trained to classify COCO objects it will almost always guess the names of COCO objects in response to the prompt "What is this object?", even if no such objects exist in the image. This can be viewed as a language bias, as has been well-studied in VQA [22, 65]. To solve this issue we re-calibrate the models output prediction by reducing the log-probability of classes that were seen in the training data when doing answer re-ranking. The down-weighting amount is selected on the validation data.

Pre-training. Recent works have shown that pre-training V+L models on large amounts of data results in large improvements [14, 47, 90]. We do not have the resources to fully-replicate these setups, but as a partial measure we pre-train GPV-T5 for 8 epochs on the CC3M dataset [68], which shows significant gains on our benchmarks. Since GPV-T5 is generative, we pre-train it by simply learning to generate the target caption rather then using a fill-in-the-blank or other more complex objectives [47, 74]. While we use much less data then state-of-the-art V+L pre-training methods, pre-training on conceptual captions does allow us to test whether GPV-T5 still benefits from web data even if exposed to a broad range of concepts during pre-training.

5. DCE Benchmark

The COCO benchmark primarily focuses on 80 object categories and is insufficient for evaluating skills on a wide range of diverse concepts. We introduce the **D**iverse **C**oncept **E**valuation (DCE) benchmark to evaluate GPV models on a large subset of the 600 OPENIMAGES cate-

Subset	Skill	Samples	Images	Categories
Val	VQA	5169	2131	295
	Localization	8756	7588	463
	Classification	9485	6770	464
	Cls-in-context	9485	6770	464
	Captioning	4500	4500	-
Test	VQA	5281	2160	307
	Localization	10586	9986	476
	Classification	10888	9161	476
	Cls-in-context	10888	9161	476
	Captioning	10600	10600	-

Table 1. **DCE Val and Test Statistics.** Since nocaps [2] annotations are hidden behind an evaluation server, we are unable to provide category counts for captioning.

gories across 5 skills: classification (Cls), classification-in-context (CiC), captioning (Cap), localization (Loc), and visual question answering (VQA). See Fig. 3 for the inputs, prompts and outputs for each task. We introduce the CiC task as a natural and unambiguous object classification task, similar to how someone may point at an object and ask what it is, and CiC provides a direct complement to localization. We source Cls, CiC and Loc samples from OpenImages, VQA samples from VisualGenome (VG), and use the nocaps [2] benchmark for Cap evaluation. To curate the DCE benchmark, we first select a set of mutually exclusive categories from OpenImages and draw samples for each of those categories according to a sampling strategy that prevents over-representation of any category while maximizing representation of "tail" categories. Note that DCE is an evaluation-only benchmark to measure understanding of GPVs on a diverse concept set across various skills, and is not accompanied by a distributionally similar training set.

Category selection. The OpenImages dataset provides a total of 600 object categories organized in a hierarchy. After removing some of the categories due to relatively high label noise, we use the remaining 492 leaf nodes in the hierarchy as our mutually exclusive set of categories.

Sampling strategy. For Cls, CiC and Loc, we randomly sample up to 25 samples from each of the selected categories. A sample for Cls/CiC is defined as any bounding box annotated with a category. For Loc, a sample is all bounding boxes in an image annotated with a category (we discard "group" annotations). For VQA, we first discard annotations exceeding 2 word answers after removing articles and tag each QA pair in VG with any of the selected categories mentioned in either the question or answer. Then, for each category, we sample up to 50 data points. Since each sample in VQA may consist of multiple categories, this strategy does result in more than 50 samples for some categories, but in practice it achieves the goal of preventing common categories from dominating the evaluation. Finally, some of the 492 categories do not have annotations in the source datasets. The final sample, image, and category counts for each skill are in Tab. 1 and category frequency histogram is shown in supplementary material.

Additional VQA annotations. VQA annotations from VG only consist of a single answer per question. For each selected VQA sample, we get 9 additional answer annotations from Amazon Mechanical Turk in accordance with standard Coco based VQA benchmarks [4,22]. Only samples where at least 3 workers agreed on an answer were retained.

6. Experiments

We train models jointly on all tasks that are supported by each GPV using Coco-based datasets. In addition, each model is also trained with and without training data from Web10k. We evaluate these models on in-domain test sets for each task as well as on the Web10k and DCE test sets.

We now summarize the tasks and training details. See Figure 3 for the inputs/outputs for each task and supplementary for more details. **VQA:** We train on the VQA v2 [22] train set and report results using the annotator-weighted metric from [22] on the VQA v2 test-dev set and DCE test set. **Captioning:** We train on Coco captioning and report CIDEr-D [77] on Coco test. DCE uses nocaps [2] for captioning, so we report CIDEr-D on nocaps in/near/out/all domain when space permits, and otherwise only report out-of-domain results since performance on novel concepts is our primary interest. **Localization:** Localization training data is built from bounding box annotations in Coco images following [25]. We report mAP on the Coco val set (since the test servers do not support this task) and the DCE test set. VL-T5 does not support this task out-of-the-box since it does not have a means to rank its input boxes, so we do not train or evaluate it for this task. **Classification:** We use the classification data from [25] and report accuracy on the Coco val set and the DCE test set. Since DCE is out-of-domain we apply the re-calibration method from Sec. 4 for GPV-T5. **Classification-in-Context:** The same as classification, except instead of cropping images the bounding box of the target object is used as an input box. Having an input box means only GPV-T5 supports this task.

Training details. We train GPV-T5 and VL-T5 for 8 epochs with a batch size of 60 and learning rate of 3e-4 that linearly warms up from 0 for 10% of the training steps and then decays to 0. We stratify the data so examples from each source are proportionally represented in each batch. Since the web data is large, we shard the data into 4 parts and use 1 shard each epoch, which results in about a third of the data in each epoch being web data. VL-T5 is initialized with the pre-trained checkpoint from [14] and GPV-T5 is initialized from our checkpoint after CC pre-training. We train GPV-1 to 40 epochs following the method of [25][2].

[2] Since [25] takes a very long time to train when using the web data (over 3 weeks), results for GPV-1 with and without web data are reported after training for 20 epochs.

		COCO					DCE					WEB10K			
Model	Web data	*VQA*	*Cap*	*Loc*	*Cls*	*CiC*	*VQA*	*Cap*	*Loc*	*Cls*	*CiC*	*All*	*Nouns*	*Verbs*	*Adj*
[a] GPV-1	no web	62.5	102.3	**73.0**	**83.6**	-	45.3	25.8	61.9	10.1	-	11.9	2.7	8.5	24.5
[b] GPV-1^{20}	no web	61.2	95.7	65.3	82.3	-	44.3	23.1	60.3	9.3	-	13.1	3.1	7.7	28.4
[c] GPV-1^{20}	with web	61.5	97.3	64.9	82.8	-	45.8	28.6	61.5	20.0	-	54.4	32.7	51.7	78.8
[d] VL-T5	no web	69.8	100.7	-	78.1	-	60.2	31.6	-	10.9	-	18.6	4.3	15.8	35.7
[e] VL-T5	with web	69.9	106.4	-	77.3	-	59.9	45.0	-	16.2	-	61.0	38.0	59.3	**85.8**
[f] GPV-T5	no web	71.1	112.1	70.9	82.2	**93.4**	60.6	65.4	74.8	36.3	43.6	22.5	3.8	23.6	39.9
[g] GPV-T5	with web	**71.4**	**113.0**	70.9	82.3	93.2	**61.1**	**72.5**	**75.9**	**45.4**	**52.2**	**62.0**	**41.7**	**60.0**	84.3

Table 2. **Concept expansion with web data.** Jointly training on WEB10K in addition to COCO shows consistent gains on DCE and WEB10K benchmarks without adversely affecting COCO performance for 3 different GPVs. DCE Cap only shows out-of-domain results from nocaps due to limited space. GPV-1^{20} refers to 20 epoch training.

		COCO-SCE												DCE							WEB10K			
		VQA			*Cap*			*Loc*			*Cls*			*VQA*	*Cap*				*Loc*	*Cls*	*All*	*Noun*	*Verb*	*Adj*
Model	Web data	*Test*	*Sn*	*Unsn*	*Test*	*Sn*	*Unsn*	*Test*	*Sn*	*Unsn*	*Test*	*Sn*	*Unsn*		*In*	*Near*	*Out*	*All*						
GPV-T5	no web	59.6	60.1	48.5	88.4	91.7	55.5	62.2	**67.2**	14.0	**73.1**	77.2	**33.9**	**46.9**	56.1	40.5	21.1	39.0	54.9	13.6	14.0	3.3	11.6	27.1
GPV-T5	with web	**59.9**	**60.3**	**49.7**	**89.2**	**92.1**	**58.0**	62.2	67.0	**14.8**	73.0	77.2	32.6	46.8	**60.8**	**47.0**	**33.4**	**46.3**	**58.7**	**26.5**	**47.0**	**25.1**	**43.0**	**73.0**

Table 3. **Concept scaling using web data: Closed world experiment.** To eliminate the effect of VinVL features and CC pretraining, we restrict GPV-T5 to COCO-SCE trained DETR features. Training jointly with WEB10K still shows massive gains on DCE and WEB10K benchmarks over training only on COCO-SCE.

Concept expansion using web data. Table 2 shows the performance of models on the three benchmarks when trained with and without WEB10K. On DCE, which contains a more diverse set of concepts than COCO, we find that all models benefit from web data and perform better on captioning and the two classification tasks (with large gains of +7.1, +9.1, +8.6 for GPV-T5). We see modest gains of +1.0 for DCE localization. VQA shows small gains, presumably because many frequent answers such as colors or numbers are common between COCO and DCE, and adding web supervision brings little benefits for such questions. Training with web data makes little difference on COCO and, unsurprisingly, leads to large gains on WEB10K test, where models achieve over 40% accuracy on nouns and 60% on verbs despite the large number of concepts. Overall, these results show multi-tasking GPVs with web data improves performance significantly on concepts unseen in supervised data without compromising in-domain performance.

Of the three GPVs we test, we find GPV-T5 to be the most effective across all three benchmarks. GPV-T5 uses less pre-training data and a simpler and cheaper pre-training strategy than VL-T5. However, it uses more powerful VinVL [90] features and benefits from classifier re-calibration (see ablations). In contrast to VL-T5, GPV-T5 can also perform classification in context and localization. In contrast to GPV-1, GPV-T5 has more powerful features and a better pre-trained language model, which help produce large gains across the benchmarks. It also trains much faster than GPV-1 since it can use pre-computed detection features (1 day on 2 GPUs vs. over 3 weeks on 4 GPUs).

Closed world evaluation of web data. Table 3 shows results for GPV-T5 when it is trained on the COCO-SCE [25] dataset, a dataset that holds out different concepts from each COCO training supervised dataset (e.g., captions that mention the word “bed” are held out from the caption training data), and then evaluates whether models can still perform well on those unseen concepts by learning about those concepts from the data in other tasks (e.g., captions with the word “bed” are in the captioning test set, and classification and localization training still include examples about beds). When GPV-T5 is trained on COCO-SCE we make two notable changes: (1) We replace VinVL features with DETR [7] features trained only on the COCO-SCE training categories (this avoids leaking detection information by VinVL’s broad category set); and (2) We do not pretrain with CC (this avoids leaking caption information from CC’s broad vocabulary). These choices severely reduce the performance of the model, but this setup serves as a closed world evaluation to determine if GPV-T5 can learn skills from COCO-SCE and concepts from WEB10K. As seen in Table 3, training with web data shows large gains across the board in this controlled experiment. In fact, we now also see gains in the unseen categories within COCO-SCE.

Ablation analysis. We perform ablations studies on GPV-T5, results are shown on the validation sets in Table 4. The model that does not use LBL scores each box using a linear classifier on top of its contextualized embedding instead. On both classification tasks and captioning, we find that web data helps with and without CC pre-training, and that removing both reduces performance dramatically (including over 30 points for captioning), showing that the two approaches are independently effective and complementary

				Coco					DCE					Web10k			
Web	CC	Cb	LBL	*VQA*	*Cap*	*Loc*	*Cls*	*CiC*	*VQA*	*Cap*	*Loc*	*Cls*	*CiC*	*All*	*Nouns*	*Verbs*	*Adj*
✓	✓	✓	✓	70.7	117.3	70.9	82.3	93.2	60.7	78.0	76.8	45.8	52.2	60.4	39.9	57.5	83.8
-	✓	✓	✓	-0.2	-1.1	0.0	-0.1	0.2	-0.5	-8.8	-1.0	-8.5	-7.4	-37.2	-35.4	-32.5	-43.8
✓	-	✓	✓	0.4	-2.4	0.1	0.5	0.1	0.8	-13.9	-0.7	-4.3	-4.5	-2.3	-3.7	-2.4	-0.9
-	-	✓	✓	0.2	-4.2	0.1	0.5	0.2	-0.2	-33.7	-4.5	-20.7	-21.1	-40.6	-37.4	-39.3	-44.9
✓	✓	-	✓	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-11.8	-12.8	0.0	0.0	0.0	0.0
✓	✓	✓	-	-0.1	-1.4	0.0	0.3	0.0	-0.2	-2.4	-1.3	-1.3	-0.7	0.1	0.2	0.7	-0.8

Table 4. **Ablating GPV-T5**. The left-most columns indicate the use of Web10k, CC pre-training, classifier re-calibration (Cb), and language-based localization (LBL) (see Sec. 4). The first row shows results for GPV-T5, and the lower rows show the differences in scores between ablated models and GPV-T5. Each component improves performance on DCE. DCE Cap shows out-of-domain results.

	imSitu (top-1 | top-5 acc.)			**VAW (mAP)**		
Model	*Test*	*Seen*	*Unsn*	*Test*	*Seen*	*Unsn*
GPV-T5	10.0 | 23.0	15.6 | 33.4	2.5 | 9.1	53.2	56.9	52.0
GPV-T5+web	16.7 | 34.7	27.5 | 54.4	2.2 | 8.3	52.4	56.2	51.3
Supervised	43.2 | 68.6	-	-	68.3	-	-

Table 5. **Zero-Shot generalization**. GPV-T5 can identify verbs and attributes on new datasets without supervision.

at helping models handle new concepts. This is also true to a more modest extent for localization. Re-calibration is critical for classification, providing a gain of up to 12 points, confirming models do tend to be overly influenced by the concept distribution observed in the training data. In-domain performance on the Coco data remains largely unchanged, which is expected since our design choices target performance on unseen concepts.

0-shot performance. We evaluate the 0-shot capabilities of GPV-T5 on an action recognition dataset (ImSitu actions [87]) and an attribute recognition dataset (VAW [61]), see Table 5. For ImSitu actions we prompt the model with "What are they doing?". GPV-T5 gets 34.7 top-5 accuracy compared to 58.6 from the benchmark authors [87] employing a supervised CNN+CRF approach and 68.6 from a recent supervised model [72] that uses a specialized mixture-kernel attention graph neural network. For verbs present in Web10k (the Seen column), Web10k training provides a significant boost (54.4 from 33.4) showing successful transfer from web images to ImSitu images. For VAW, we prompt the model with yes/no questions (e.g., "Is this object pink?") along with the target object's bounding box to get per-box multi-label attribute results. Even without Web10k, 0-shot GPV-T5 performs surprisingly well (53.2 vs. 68.3 mAP for a fully supervised model [61]), likely because the model already learns these attributes from VinVL, CC, VQA, and Captioning training data.

Human object interaction. To demonstrate the flexibility of GPV-T5, we also employ it for human-object interaction detection [8] using the two-stage procedure described in Sec. 4. We fine-tune GPV-T5 on the HICO-Det train set for four epochs (see supplemental for details). GPV-T5 gets an AP of 20.6 on the HICO-Det benchmark, which is comparable to a number of other approaches (17.2 [26], 19.8 [75], 20.8 [93], 21.8 [18]). Although recent models [36, 89, 94] show results up to 32.1 mAP [89], they require highly specialized architectures requiring up to 5 output heads (e.g. for decoding human+object boxes, interaction score, and object and interaction categories), well crafted losses (e.g. Hungarian HOI instance matching objectives), and custom post-processing steps (e.g pairwise non-maximum suppression). GPV-T5's flexibility allows us to get reasonable results by side-stepping complex model design with simple chained inference.

Qualitative results. See Supp. for qualitative results for GPV-T5 on the three benchmarks.

7. Discussion

Limitations. GPV-T5 achieves transfer of concepts from web data to skills, but our results indicate that more work is needed, particularly for tasks like VQA or localization, e.g., through new architectures or training protocols. GPV-T5 supports a wide range of tasks, but the ability to handle more modalities (e.g., video) and outputs (e.g., segmentation) would enable even more. Recent work shows promise in this regard [30], and raises the potential of transferring concepts from web data to an even wider range of tasks.

Potential negative impact. We employ several measures to ensure Web10k is clean including the "isFamilyFriendly" filter on Bing, removing inappropriate words per a popular blocklist [1], and conducting manual spot checks. However, the entire dataset has not been human-curated, so we cannot guarantee it is free from objectionable imagery. It is important to be aware that search results are known to reflect human biases and stereotypes [34, 58], for example, most of our images for "soccer player" are of men. Coco, our main source of supervision, also suffers from these kinds of biases [91] so we do not recommend using the models in this paper in production settings.

Conclusion. As the vision community builds progressively more general models, identifying efficient ways of learning a large variety of skills and concepts is of prime importance. Our work revisits the idea of webly-supervised learning in the context of GPVs and shows that learning skills from task-specific datasets and concepts from the web is an efficient and inexpensive option for concept expansion.

References

[1] Offensive/profane word list. http://www.cs.cmu.edu/biglou/resources/bad-words.txt. 8

[2] Harsh Agrawal, Karan Desai, Yufei Wang, Xinlei Chen, Rishabh Jain, Mark Johnson, Dhruv Batra, Devi Parikh, Stefan Lee, and Peter Anderson. nocaps: novel object captioning at scale. *International Conference on Computer Vision*, pages 8947–8956, 2019. 3, 6, 2

[3] Peter Anderson, X. He, C. Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6077–6086, 2018. 3

[4] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. VQA: Visual question answering. In *ICCV*, 2015. 6

[5] T. Brown, B. Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, G. Krüger, T. Henighan, R. Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, E. Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, J. Clark, Christopher Berner, Sam McCandlish, A. Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. *ArXiv*, abs/2005.14165, 2020. 2

[6] Marc Brysbaert, Amy Warriner, and Victor Kuperman. Concreteness ratings for 40 thousand generally known english word lemmas. *Behavior research methods*, 46, 10 2013. 3

[7] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. *ECCV*, abs/2005.12872, 2020. 2, 5, 7

[8] Yu-Wei Chao, Yunfan Liu, Xieyang Liu, Huayi Zeng, and Jia Deng. Learning to detect human-object interactions. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*, 2018. 5, 8, 6

[9] Yu-Wei Chao, Zhan Wang, Yugeng He, Jiaxuan Wang, and Jia Deng. HICO: A benchmark for recognizing human-object interactions in images. In *Proceedings of the IEEE International Conference on Computer Vision*, 2015. 4, 5

[10] Huizhong Chen, Andrew C. Gallagher, and Bernd Girod. Describing clothing by semantic attributes. In *ECCV*, 2012. 4

[11] Xinlei Chen and Abhinav Gupta. Webly supervised learning of convolutional networks. In *ICCV*, 2015. 3

[12] Xinlei Chen, Abhinav Shrivastava, and Abhinav Gupta. NEIL: Extracting visual knowledge from web data. In *ICCV*, 2013. 3

[13] Yen-Chun Chen, Linjie Li, Licheng Yu, A. E. Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and J. Liu. Uniter: Learning universal image-text representations. *ArXiv*, abs/1909.11740, 2019. 3

[14] Jaemin Cho, Jie Lei, Hao Tan, and Mohit Bansal. Unifying vision-and-language tasks via text generation. *arXiv preprint arXiv:2102.02779*, 2021. 1, 3, 5, 6

[15] Santosh Divvala, Ali Farhadi, and Carlos Guestrin. Learning everything about anything: Webly-supervised visual concept learning. In *CVPR*, 2014. 3

[16] Wei Dong, Richard Socher, Li Li-Jia, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009. 3

[17] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, abs/2010.11929, 2021. 2

[18] Haoshu Fang, Yichen Xie, Dian Shao, and Cewu Lu. Dirv: Dense interaction region voting for end-to-end human-object interaction detection. In *AAAI*, 2021. 8

[19] Ali Farhadi, Ian Endres, Derek Hoiem, and David Alexander Forsyth. Describing objects by their attributes. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1778–1785, 2009. 4

[20] Rob Fergus, Li Fei-Fei, Pietro Perona, and Andrew Zisserman. Learning object categories from google's image search. *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, 2:1816–1823 Vol. 2, 2005. 3

[21] Eren Golge and Pinar Duygulu Sahin. Conceptmap: Mining noisy web data for concept learning. In *ECCV*, 2014. 3

[22] 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. 5, 6

[23] Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. Open-vocabulary object detection via vision and language knowledge distillation, 2021. 3

[24] Sheng Guo, Weilin Huang, Haozhi Zhang, Chenfan Zhuang, Dengke Dong, Matthew R. Scott, and Dinglong Huang. Curriculumnet: Weakly supervised learning from large-scale web images. *ArXiv*, abs/1808.01097, 2018. 3

[25] Tanmay Gupta, Amita Kamath, Aniruddha Kembhavi, and Derek Hoiem. Towards general purpose vision systems. *arXiv preprint arXiv:2104.00743*, 2021. 1, 2, 3, 5, 6, 7

[26] Tanmay Gupta, Alexander G. Schwing, and Derek Hoiem. No-frills human-object interaction detection: Factorization, layout encodings, and training techniques. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9676–9684, 2019. 8

[27] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. Mask R-CNN. In *ICCV*, 2017. 2

[28] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1

[29] Judy Hoffman, Sergio Guadarrama, Eric Tzeng, Ronghang Hu, Jeff Donahue, Ross B. Girshick, Trevor Darrell, and Kate Saenko. Lsda: Large scale detection through adaptation. In *NIPS*, 2014. 3

[30] Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu, David Ding, Skanda Koppula, Andrew Brock, Evan Shelhamer, Olivier J. H'enaff, Matthew M. Botvinick, Andrew Zisserman, Oriol Vinyals, and João Carreira. Perceiver io: A general architecture for structured inputs & outputs. *ArXiv*, abs/2107.14795, 2021. 3, 8

[31] Andrew Jaegle, Felix Gimeno, Andrew Brock, Andrew Zisserman, Oriol Vinyals, and João Carreira. Perceiver: General perception with iterative attention. In *ICML*, 2021. 1, 3

[32] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V. Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *ICML*, 2021. 3

[33] Bin Jin, Maria V. Ortiz Segovia, and Sabine Süsstrunk. Webly supervised semantic segmentation. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1705–1714, 2017. 3

[34] Matthew Kay, Cynthia Matuszek, and Sean A Munson. Unequal representation and gender stereotypes in image search results for occupations. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 3819–3828, 2015. 8

[35] Alex Kendall, Matthew Koichi Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 2938–2946, 2015. 1

[36] Bumsoo Kim, Junhyun Lee, Jaewoo Kang, Eun-Sol Kim, and Hyunwoo J Kim. Hotr: End-to-end human-object interaction detection with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 8

[37] Wonjae Kim, Bokyung Son, and Ildoo Kim. Vilt: Vision-and-language transformer without convolution or region supervision. *ArXiv*, abs/2102.03334, 2021. 2

[38] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[39] Jonathan Krause, Benjamin Sapp, Andrew G. Howard, Howard Zhou, Alexander Toshev, Tom Duerig, James Philbin, and Li Fei-Fei. The unreasonable effectiveness of noisy data for fine-grained recognition. *ArXiv*, abs/1511.06789, 2016. 3

[40] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, Michael Bernstein, and Li Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *IJCV*, 123:32–73, 2017. 2, 4, 6

[41] Neeraj Kumar, Alexander C. Berg, Peter N. Belhumeur, and Shree K. Nayar. Attribute and simile classifiers for face verification. *2009 IEEE 12th International Conference on Computer Vision*, pages 365–372, 2009. 4

[42] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper R. R. Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Tom Duerig, and Vittorio Ferrari. The Open Images Dataset V4: Unified image classification, object detection, and visual relationship detection at scale. *arXiv:1811.00982*, 2018. 6

[43] Christoph H. Lampert, Hannes Nickisch, and Stefan Harmeling. Learning to detect unseen object classes by between-class attribute transfer. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 951–958, 2009. 4

[44] Liunian Harold Li, Mark Yatskar, Da Yin, C. Hsieh, and Kai-Wei Chang. Visualbert: A simple and performant baseline for vision and language. *ArXiv*, abs/1908.03557, 2019. 3

[45] Li-Jia Li and Li Fei-Fei. Optimol: Automatic online picture collection via incremental model learning. *International Journal of Computer Vision*, 88:147–168, 2007. 3

[46] Quannan Li, Jiajun Wu, and Zhuowen Tu. Harvesting mid-level visual concepts from large-scale internet images. *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 851–858, 2013. 3

[47] Xiujun Li, Xi Yin, C. Li, X. Hu, Pengchuan Zhang, Lei Zhang, Longguang Wang, H. Hu, Li Dong, Furu Wei, Yejin Choi, and Jianfeng Gao. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *ECCV*, 2020. 3, 5

[48] Kongming Liang, Yuhong Guo, Hong Chang, and Xilin Chen. Visual relationship detection with deep structural ranking. In *AAAI*, 2018. 4

[49] 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 *ECCV*, 2014. 2, 6

[50] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Ching-Feng Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *ICCV*, abs/2103.14030, 2021. 2

[51] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *NeurIPS*, 2019. 3

[52] Jiasen Lu, Vedanuj Goswami, Marcus Rohrbach, D. Parikh, and Stefan Lee. 12-in-1: Multi-task vision and language representation learning. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10434–10443, 2020. 2

[53] Jiasen Lu, Vedanuj Goswami, Marcus Rohrbach, Devi Parikh, and Stefan Lee. 12-in-1: Multi-task vision and language representation learning. In *CVPR*, 2020. 3

[54] Ao Luo, Xin Li, Fan Yang, Zhicheng Jiao, and Hong Cheng. Webly-supervised learning for salient object detection. *Pattern Recognit.*, 103:107308, 2020. 3

[55] B. McCann, N. Keskar, Caiming Xiong, and R. Socher. The natural language decathlon: Multitask learning as question answering. *ArXiv*, abs/1806.08730, 2018. 2

[56] Li Niu, Qingtao Tang, Ashok Veeraraghavan, and Ashutosh Sabharwal. Learning from noisy web data with category-level supervision. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7689–7698, 2018. 3

[57] Li Niu, Ashok Veeraraghavan, and Ashutosh Sabharwal. Webly supervised learning meets zero-shot learning: A hybrid approach for fine-grained classification. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7171–7180, 2018. 3

[58] Jahna Otterbacher, Jo Bates, and Paul Clough. Competent men and warm women: Gender stereotypes and backlash in image search results. In *Proceedings of the 2017 chi conference on human factors in computing systems*, pages 6620–6631, 2017. 8

[59] Devi Parikh and Kristen Grauman. Relative attributes. *2011 International Conference on Computer Vision*, pages 503–510, 2011. 4

[60] Genevieve Patterson and James Hays. Sun attribute database: Discovering, annotating, and recognizing scene attributes. *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2751–2758, 2012. 4

[61] Khoi Pham, Kushal Kafle, Zhe Lin, Zhihong Ding, Scott Cohen, Quan Tran, and Abhinav Shrivastava. Learning to predict visual attributes in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13018–13028, June 2021. 8

[62] 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, 2021. 3

[63] 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 *ICML*, 2021. 3

[64] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, W. Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67, 2020. 2, 3

[65] Sainandan Ramakrishnan, Aishwarya Agrawal, and Stefan Lee. Overcoming language priors in visual question answering with adversarial regularization. *arXiv preprint arXiv:1810.03649*, 2018. 5

[66] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, 2016. 1

[67] Joseph Redmon and Ali Farhadi. Yolo9000: Better, faster, stronger. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6517–6525, 2017. 3

[68] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *ACL*, 2018. 3, 4, 5, 6

[69] Sheng Shen, Liunian Harold Li, Hao Tan, Mohit Bansal, Anna Rohrbach, Kai-Wei Chang, Zhewei Yao, and Kurt Keutzer. How much can clip benefit vision-and-language tasks? *ArXiv*, abs/2107.06383, 2021. 3

[70] Tong Shen, Guosheng Lin, Chunhua Shen, and Ian D. Reid. Bootstrapping the performance of webly supervised semantic segmentation. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1363–1371, 2018. 3

[71] Yunhang Shen, Rongrong Ji, Zhiwei Chen, Xiaopeng Hong, Feng Zheng, Jianzhuang Liu, Mingliang Xu, and Qi Tian. Noise-aware fully webly supervised object detection. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11323–11332, 2020. 3

[72] Mohammed Suhail and Leonid Sigal. Mixture-kernel graph attention network for situation recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10363–10372, 2019. 8

[73] Guolei Sun, Wenguan Wang, Jifeng Dai, and Luc Van Gool. Mining cross-image semantics for weakly supervised semantic segmentation. In *ECCV*, 2020. 3

[74] Hao Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. In *EMNLP/IJCNLP*, 2019. 3, 5

[75] Oytun Ulutan, A S M Iftekhar, and B. S. Manjunath. Vsgnet: Spatial attention network for detecting human object interactions using graph convolutions. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13614–13623, 2020. 8

[76] Laurens van der Maaten and Geoffrey E. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605, 2008. 4

[77] Ramakrishna Vedantam, C. L. Zitnick, and D. Parikh. Cider: Consensus-based image description evaluation. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4566–4575, 2015. 6

[78] Sudheendra Vijayanarasimhan and Kristen Grauman. Keywords to visual categories: Multiple-instance learning forweakly supervised object categorization. *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008. 3

[79] Shuo Wang, Jungseock Joo, Yizhou Wang, and Song-Chun Zhu. Weakly supervised learning for attribute localization in outdoor scenes. *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3111–3118, 2013. 4

[80] Shufan Wang, Laure Thompson, and Mohit Iyyer. Phrase-bert: Improved phrase embeddings from bert with an application to corpus exploration. In *EMNLP*, 2021. 4

[81] Xin-Jing Wang, Lei Zhang, Xirong Li, and Wei-Ying Ma. Annotating images by mining image search results. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30:1919–1932, 2008. 3

[82] Spencer Whitehead, Hui Wu, Heng Ji, Rogério Schmidt Feris, Kate Saenko, and Uiuc MIT-IBM. Separating skills and concepts for novel visual question answering. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5628–5637, 2021. 3

[83] Zhonghua Wu, Qingyi Tao, Guosheng Lin, and Jianfei Cai. Exploring bottom-up and top-down cues with attentive learning for webly supervised object detection. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12933–12942, 2020. 3

[84] Haiyang Xu, Ming Yan, Chenliang Li, Bin Bi, Songfang Huang, Wenming Xiao, and Fei Huang. E2e-vlp: End-to-end vision-language pre-training enhanced by visual learning. 2021. 3

[85] Jingkang Yang, Litong Feng, Weirong Chen, Xiaopeng Yan, Huabin Zheng, Ping Luo, and Wayne Zhang. Webly super-

vised image classification with self-contained confidence. In *ECCV*, 2020. 3

[86] Mark Yatskar, Luke Zettlemoyer, and Ali Farhadi. Situation recognition: Visual semantic role labeling for image understanding. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5534–5542, 2016. 4

[87] Mark Yatskar, Luke Zettlemoyer, and Ali Farhadi. Situation recognition: Visual semantic role labeling for image understanding. In *Conference on Computer Vision and Pattern Recognition*, 2016. 8

[88] Alireza Zareian, Kevin Dela Rosa, Derek Hao Hu, and Shih-Fu Chang. Open-vocabulary object detection using captions. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14388–14397, 2021. 3

[89] Aixi Zhang, Yue Liao, Si Liu, Miao Lu, Yongliang Wang, Chen Gao, and Xiaobo Li. Mining the benefits of two-stage and one-stage hoi detection. *arXiv preprint arXiv:2108.05077*, 2021. 8

[90] Pengchuan Zhang, Xiujun Li, X. Hu, Jianwei Yang, L. Zhang, Li-Juan Wang, Yejin Choi, and Jianfeng Gao. Vinvl: Making visual representations matter in vision-language models. *ArXiv*, abs/2101.00529, 2021. 3, 4, 5, 7

[91] Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. Men also like shopping: Reducing gender bias amplification using corpus-level constraints. In *EMNLP*, 2017. 8

[92] Wenbo Zheng, Lan Yan, Chao Gou, and Feiyue Wang. Webly supervised knowledge embedding model for visual reasoning. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12442–12451, 2020. 3

[93] Xubin Zhong, Changxing Ding, Xian Qu, and Dacheng Tao. Polysemy deciphering network for human-object interaction detection. In *ECCV*, 2020. 8

[94] Cheng Zou, Bohan Wang, Yue Hu, Junqi Liu, Qian Wu, Yu Zhao, Boxun Li, Chenguang Zhang, Chi Zhang, Yichen Wei, et al. End-to-end human object interaction detection with hoi transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 8

Appendix -

Webly Supervised Concept Expansion for General Purpose Vision Models

The appendix includes the following sections:

- Sec 1 - Qualitative results from GPV-T5
- Sec 2 - Classification re-calibration analysis
- Sec 3 - Web10k Questions
- Sec 4 - DCE sampling details
- Sec 5 - GPV-T5 efficiency metrics
- Sec 6 - Experimental details
- Sec 7 - Human-object interaction details
- Sec 8 - License information

1. Qualitative results from GPV-T5

Qualitative results from GPV-T5 are shown in Figure 1. Despite the presence of concepts that are not annotated in Coco (e.g, “Caterpillar”, “Lifejackets”, “Willow”) GPV-T5 is able to successfully perform classification, localization, captioning, and visual questioning answering. Visualizations of predictions from GPV-T5 on *randomly selected* examples from the Coco, DCE, and Web10k datasets can be found in additional files in the supplementary.

Figure 2 contrasts the predictions of GPV-T5 when trained with and without Web10k. In many of these examples, the model trained without web data generates Coco concepts even when they are not present in the image (e.g., writing a caption about a giraffe for a picture of a jaguar, a brown-and-white bear for a red panda, or classifying a monkey as a bear), while the model trained on web data is able to name the new concepts correctly. For localization, we observe cases where the model trained without Web10k struggles on new concepts (e.g., the without web model focuses on humans for the class “balance beam”) while the model trained with Web10k can localize them accurately.

2. Classification re-calibration analysis

In this section, we analyze the classification re-calibration method from Sec. 4. Table 1 shows a breakdown of how GPV-T5 behaves on DCE classification with and without re-calibration. Without re-calibration GPV-T5 predicts a Coco category for 56% of CiC examples and 65.7% of the CLS examples, even though only 14% of these examples have a labels that is a Coco category, showing that the model has a strong bias towards these categories. Adding re-calibration mostly mitigates this bias and significantly boosts performance on non-Coco categories. It comes at the cost of some performance on examples that belong to Coco categories, but those examples are only a small portion of the data so performance is increased by 12 points overall. These results show re-calibration is an important component to allowing models to transfer concepts learned from non-classification data to the classification skill. Qualitative examples are shown in Figure 3.

Task	Cb	Acc.	Coco Acc.	Other Acc.	Coco Ans.
CiC	-	39.4	92.0	30.8	56.4
CiC	✓	52.2	77.5	48.1	19.7
CLS	-	34.0	85.7	25.5	65.7
CLS	✓	45.8	69.9	41.9	24.2

Table 1. **GPV-T5 accuracy on DCE classification with and without classifier re-calibration (Cb)**. The Acc. column shows overall accuracy, Coco Acc. shows accuracy on examples with labels in the 80 Coco categories, Other Acc. shows accuracy on other examples, and Coco Ans. shows how often the model predicts a Coco category.

3. Web10k Questions

In this section, we provide more detail about how we construct question-answer pairs from the web search data. For each query-image pair, we construct a question that is answered by the noun from the query. For example, the question “What entity is this?” with the answer “dog” for the query “brown dog”. For queries that contain a verb, we construct two additional questions that are answered by the verb, one that specifies the noun and one that does not. For example, “What action is happening?”, and “What is the dog doing?” with the answer “running”, for the query “dog running”. For queries that contain adjectives, we similarly construct two questions that are answered by the adjective, one that specifies the noun and one that does not. We additionally manually map the adjectives to adjective types (e.g., “color” for “red”) and specify the adjective type in the question. For example, “What is the color of this object?” and “What is the color of this dog?” with the answer “brown”, for the query “brown dog”. The mapping to adjective types is important to avoid generic questions like “What attributes does this object have?” that will have many possible correct answers. During evaluation, we compute the average accuracy on questions where the is answer is a noun, verb or adjective, and report the macro-average of those results to get an overall accuracy number.

The questions themselves are generated by a templating system to increase their linguistic diversity. Table 2 shows the templates we use. For a given query and question type we use these templates to generate a large number of pos-

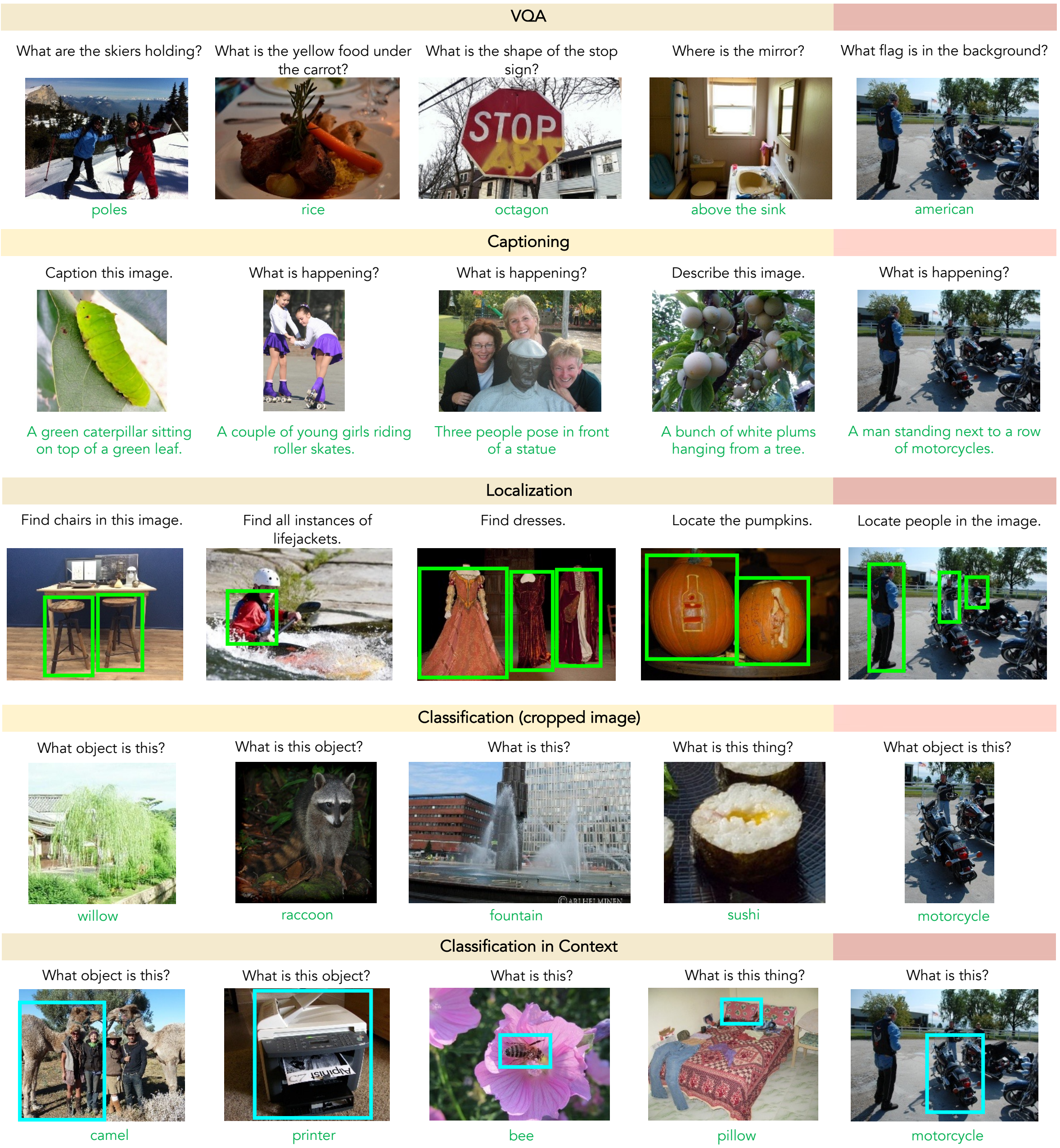

VQA
What are the skiers holding?
poles
What is the yellow food under the carrot?
rice
What is the shape of the stop sign?
octagon
Where is the mirror?
above the sink
What flag is in the background?
american
Captioning
Caption this image.
A green caterpillar sitting on top of a green leaf.
What is happening?
A couple of young girls riding roller skates.
What is happening?
Three people pose in front of a statue
Describe this image.
A bunch of white plums hanging from a tree.
What is happening?
A man standing next to a row of motorcycles.
Localization
Find chairs in this image.
Find all instances of lifejackets.
Find dresses.
Locate the pumpkins.
Locate people in the image.
Classification (cropped image)
What object is this?
willow
What is this object?
raccoon
What is this?
fountain
What is this thing?
sushi
What object is this?
motorcycle
Classification in Context
What object is this?
camel
What is this object?
printer
What is this?
bee
What is this thing?
pillow
What is this?
motorcycle

Figure 1. **Qualitative examples for GPV-T5**. Examples are from DCE val, except for the last image in each row, which comes from COCO val. GPV-T5 is able to use concepts that do not appear in the COCO training data across all five skills.

sible questions, and then select one at random to use as a prompt for the model.

Additional question types are possible. For example, contrastive questions like “Is this sloth swimming or climbing?”, or questions that specify hypernyms of the answer (obtained from sources such as WordNet) like “What kind of reptile is this?”. We leave the generation of such questions, as well as their impact on knowledge transfer of concepts between skills, to future work.

4. DCE sampling details

Fig. 4 shows the number of categories with various frequencies of occurrence in the DCE val and test sets. Since NOCAPS [2] annotations are hidden behind an evaluation server, we are unable to provide category counts for captioning. Note that VQA has fewer concepts for higher fre-

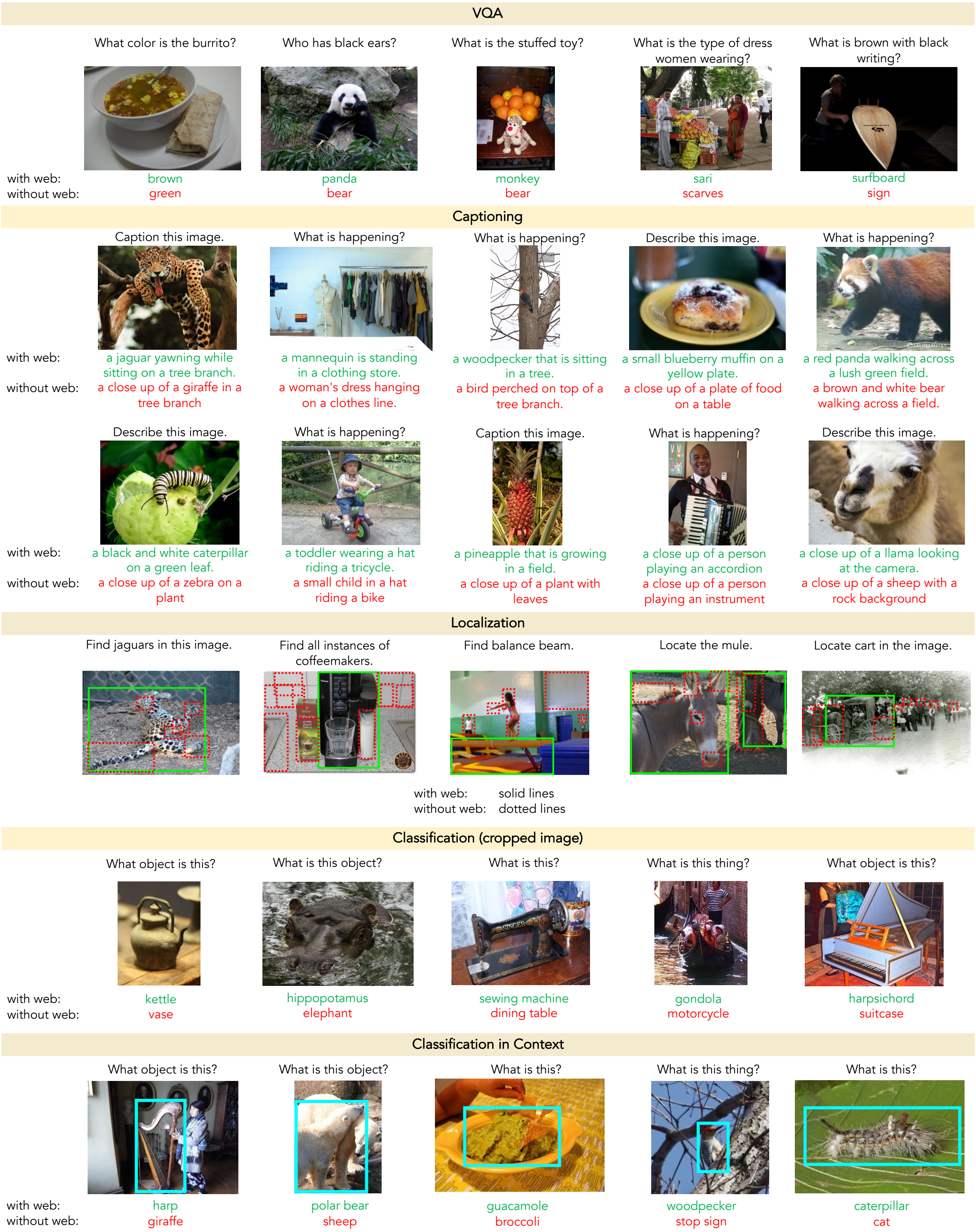

VQA
What color is the burrito?
Who has black ears?
What is the stuffed toy?
What is the type of dress women wearing?
What is brown with black writing?
with web:
without web:
brown
green
panda
bear
monkey
bear
sari
scarves
surfboard
sign
Captioning
Caption this image.
What is happening?
What is happening?
Describe this image.
What is happening?
with web:
without web:
a jaguar yawning while sitting on a tree branch.
a close up of a giraffe in a tree branch
a mannequin is standing in a clothing store.
a woman's dress hanging on a clothes line.
a woodpecker that is sitting in a tree.
a bird perched on top of a tree branch.
a small blueberry muffin on a yellow plate.
a close up of a plate of food on a table
a red panda walking across a lush green field.
a brown and white bear walking across a field.
Describe this image.
What is happening?
Caption this image.
What is happening?
Describe this image.
with web:
without web:
a black and white caterpillar on a green leaf.
a close up of a zebra on a plant
a toddler wearing a hat riding a tricycle.
a small child in a hat riding a bike
a pineapple that is growing in a field.
a close up of a plant with leaves
a close up of a person playing an accordion
a close up of a person playing an instrument
a close up of a llama looking at the camera.
a close up of a sheep with a rock background
Localization
Find jaguars in this image.
Find all instances of coffeemakers.
Find balance beam.
Locate the mule.
Locate cart in the image.
with web: solid lines
without web: dotted lines
Classification (cropped image)
What object is this?
What is this object?
What is this?
What is this thing?
What object is this?
with web:
without web:
kettle
vase
hippopotamus
elephant
sewing machine
dining table
gondola
motorcycle
harpsichord
suitcase
Classification in Context
What object is this?
What is this object?
What is this?
What is this thing?
What is this?
with web:
without web:
harp
giraffe
polar bear
sheep
guacamole
broccoli
woodpecker
stop sign
caterpillar
cat

Figure 2. **Qualitative Examples: GPV-T5 on DCE, with and without training on WEB10K**. The use of WEB10K allows GPV-T5 to understand more concepts across all skills, especially for rare concepts such as "red panda" (captioning upper right).

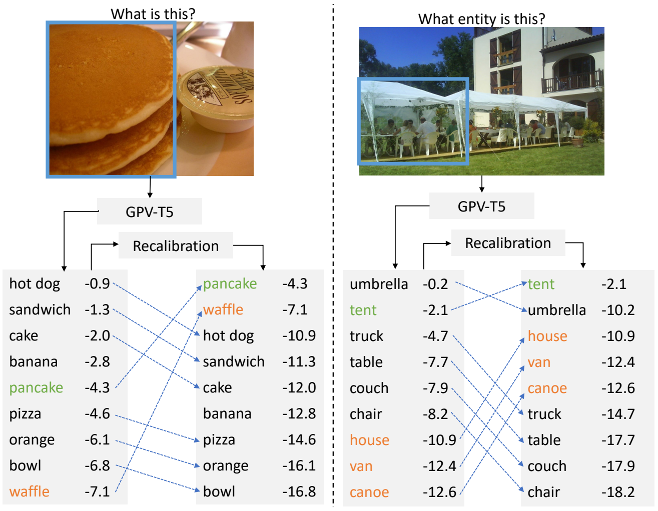


Figure 3. **Qualitative examples of re-calibration**. This figure shows two CiC examples, where the left tables show GPV-T5's top 9 predictions and log-probability scores, and the right table shows how the scores and rankings change after re-calibration. The model has a strong preference for answers seen in the COCO classification data (black), resulting in the model ranking COCO classes that are vaguely visually similar to the image over the correct class (green). Re-calibration increases the relative score of the non-COCO answers (green if correct, orange otherwise) allowing the model to get these examples correct.

quencies than localization and captioning because of a lack of a sufficient number of question-answer annotations that mention many of the OPENIMAGES categories selected for DCE.

VQA sampling strategy. Co-occurrence of concepts in questions and answers, makes the sampling strategy for VQA more nuanced than that followed for Cls, CiC, and Loc. Specifically, we iterate over the categories selected for DCE and randomly sample up to 50 samples for each category. Unlike Cls/CiC and Loc, each sample in VQA may consist of multiple categories. If k samples have already been sampled for the i^{th} category in the selected category list due to co-occurrence with previous $i-1$ categories, we only sample $\max(0, 50-k)$ samples for the i^{th} category. This allows the "tail" categories from the original dataset to be maximally sampled, while "head" categories are skipped if already sufficiently represented in the annotations sampled thus far.

Answer Type	**Prompts**
Noun	What is DT OBJ? What OBJ is this? What OBJ is that? Classify DT OBJ. Specify DT OBJ. Name DT OBJ.
Adjective	WH ADJ_TYPE is DT OBJ? What is the ADJ_TYPE of DT OBJ? CMD the ADJ_TYPE of DT OBJ.
Verb	What is DT OBJ doing? What action is DT OBJ taking? What action is DT OBJ performing? What action is DT OBJ carrying out? What action is DT OBJ doing? What activity is DT OBJ doing? CMD the action being taken by DT OBJ. CMD the activity DT OBJ is doing. CMD what DT OBJ is doing. What is being done? WH action is being done? WH activity is being done? WH activity is this? WH action is being taken? CMD the activity being done. CMD the action being done. CMD the action being taken.

DT → the, this, that
OBJ → entity, object
WH → What, Which
CMD → Describe, State, Specify, Name

Table 2. **Templates for generating web prompts**. The three sections of the tables show question templates for questions that have a noun, verb, or adjective answer. These templates are expanded by substituting the all-caps words for any one of the substitute words specified below the table, except ADJ_TYPE which is replaced by the type of the adjective for questions with adjective answers. For verb and adjective questions where the object is specified, OBJ is replaced by the noun instead, and verb templates that do not contain OBJ are not used.

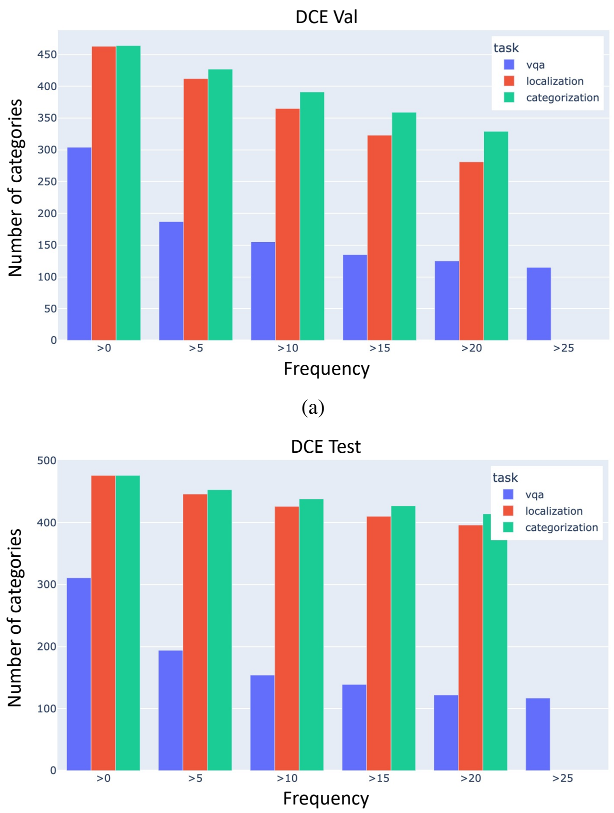
Pre.	Params	VQA	Cap	Loc	CLS	CiC
✓	224M	4.68G	6.31G	25.1G	2.63G	4.73G
-	370M	7.35T	7.38T	7.64T	6.62T	7.30T

Table 3. **Number of parameters and FLOPs in GPV-T5**. Results are shown for both when the image features are pre-computed (top), and when they have to be generated from scratch (bottom).

5. GPV-T5 efficiency metrics

We report efficiency metrics on GPV-T5 when features must be extracted from the input image from scratch using VinVL, and for when those features are assumed to have been precomputed. We report parameter count and the number of floating-point operations (FLOPs). Since the number of FLOPs depends on the length of the input, the length of the target text, and the number of regions in the image, we report the average number of FLOPs needed to process a single example on 100 random examples from the training sets for each task. We compute FLOPs using a pytorch profiler [1] while computing the loss with a single forward pass of

[1] https://github.com/facebookresearch/fvcore/blob/main/docs/flop_count.md



(a)

(b)

Figure 4. **DCE val and test set category frequencies.** Bars at $> x$ indicate the number of categories with at least x samples per category for each DCE skill with publicly available annotations. DCE expands the scope of concept evaluation across skills beyond COCO's 80 concepts and maximizes representation of a large subset of mutually exclusive concepts in OPENIMAGES while avoiding over-representation of "head" concepts (e.g. "man", "woman").

the model. Results are shown in Table 3. We find captioning is slow due to the long output sequences, classification is fast because the output text is short and there tends to be fewer objects in the cropped classification images, and detection requires generating per-box outputs so it requires the most compute. If computing the features from scratch, the computational cost is dominated by VinVL, which requires running a X152-FPN backbone and computing features for a large number of proposal regions [90].

6. Experimental Details

Here we give a more detailed account of how the models are trained. We train GPV-T5 and VL-T5 using the Adam optimizer [38] with a batch size of 60 and learning rate of 3e-4, β_1 of 0.9, β_1 of 0.999, ϵ of 1e-8, and a weight decay of 1e-4. The learning rate linearly warms up from 0 for 10 of the training steps and then linear decays back to 0 at the end of training. The web data is sharded into 4 parts, and a different part of used for each epoch for the first four epochs. Then the data is re-sharded into 4 new parts for the final 4 epochs. The data is stratified so that the 6 supervised datasets (VQA, Cap, Loc, CLS, CiC and the current web shard) are represented in approximately the same proportion in each batch. During training, we use the cross-entropy loss of generating the output text for all tasks besides localization. For localization, we compute relevance scores for each box following the process in Sec. 4 and then train using the Hungarian-matching loss from DETR [7] with two classes (one class for relevant and one for irrelevant) following [25]. We compute the scores on the in-domain validation sets each epoch, and use the checkpoint with the highest average score across all validation tasks. We experimented with using different learning rates for VL-T5 but found it had little impact on performance, so used the same learning rates for both models. We use the prompts created by [25] for CLS, Loc and Cap, and from our questions template for WEB10K (See Sec. 3). For CiC we use the CLS prompts. During testing, we generate text using beam search with a beam size of 20, except for classification on in which case we use the ranking approach from Sec. 4.

7. Human object interactions details

In this section, we provide more details about how GPV-T5 is trained to perform human-object interaction. Both stages of the two-pass process from Sec. 4 are trained using the HOI-Det training set [8]. The first pass requires the model to locate person bounding boxes in the image, GPV-T5 is trained to do this by using localization examples constructed from the HOI annotations. In particular, we build examples by gathering all person-boxes in the annotations for an image and then pruning duplicate boxes by applying non-maximum suppression with a threshold of 0.7. The remaining boxes serve as ground truth for localization examples with the prompt "Locate the people".

The second pass requires the model to identify object interactions given a person box. GPV-T5 is trained using the same de-duplicated person boxes from the HOI annotations. For each such person box, the input to the model is the image with the prompt "What is this person doing?" and the input query box set to be the person box. Target outputs are built by gathering all HOI annotations for that input person box (annotations with person boxes that were pruned during de-duplication are mapped to the person box with the highest IoU overlap). This results in a set of object boxes labeled with HOI classes for each person box. Those object boxes are aligned with the boxes found by the object detector by finding the box with the highest IoU overlap with each ground truth object box. During training, if no box from the object detector has at least a 0.5 overlap with an object box, we manually add that object box to the regions extracted by the detector so we can still train on it. The model is trained to generate a text description of the HOI class for each box that was aligned with a ground truth box (e.g., "riding the horse" for the HOI class riding+horse), or the text "no interaction" for any box that was not aligned with a ground truth object. In practice, we only train on a randomly selected half of the "no interaction" boxes to reduce computational expense. If an object box is aligned to multiple ground truth boxes, and therefore has multiple HOI class labels, we train the model to generate all such labels with a high probability.

We train the model with the hyper-parameters specified in Sec. 6, but for 4 epochs with a batch of 48 and a learning rate of 1e-4. Since this task is intended as a demonstration, we did not spend a lot of time optimizing this process and think it could be further improved with additional effort.

To evaluate the model, we first find boxes the model identifies from the prompt "Locate the people" with a score of over 0.5. Then for each such box, for each object box detected by the object detector, and for each HOI class, we score the box pair and class with the log-probability of generating the class label text from the object box when the person box is used as the input query box. In practice, for a given person box, we prune object boxes that generate the text "no interaction" with a high probability so we do not have to score a generation for every class label with that box-pair. These scores are finally used to compute the average precision metric from [8].

Find HOIs for an image requires one forward pass with the encoder for each person box, then one forward pass with the decoder for each person box/object box pair to compute the "no interaction" probability, and then another forward pass with the decoder for each person box, non-pruned object box, and class label to get the class scores. This is made affordable by the fact the class labels are short, and we are able to label the 10k test set in about an hour using a single Quadro RTX 8000 GPU (after the VinVL image features have been precomputed).

8. License information

Licenses for datasets used in this work are:

- CONCEPTUAL CAPTIONS [68]: A custom open-source license [2]
- NOCAPS [2]: Creative Commons Attribution 2.0 Generic
- VISUALGENOME [40]: Creative Commons Attribution 4.0 International License
- VQA V2 [22]: Commons Attribution 4.0 International License
- COCO [49]: Creative Commons Attribution 4.0 License
- OPENIMAGES [42]: Creative Commons Attribution 4.0 License

Licenses for code libraries used in this work are attached separately in licenses.txt. Our code additionally uses elements from the GPV-1 code base[3] [25] (Apache 2.0 license).

[2] https://github.com/google-research-datasets/conceptual-captions/blob/master/LICENSE

[3] https://github.com/allenai/gpv