

Webly Supervised Concept Expansion for General Purpose Vision Models

Amita Kamath^{*1}, Christopher Clark^{*1}, Tanmay Gupta^{*1}, Eric Kolve¹, Derek Hoiem², and Aniruddha Kembhavi¹

¹ Allen Institute for Artificial Intelligence

² University of Illinois at Urbana-Champaign

Abstract. General Purpose Vision (GPV) systems 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 supervised datasets, learn concepts from web image search, 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 and VL-T5) on 3 benchmarks: 5 COCO-based datasets (80 primary concepts), a newly curated series of 5 datasets based on the OpenImages and VisualGenome repositories (\sim 500 concepts), and the Web-derived dataset (10k+ concepts). We also propose a new architecture, GPV-2 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 detection. GPV-2 benefits hugely from web data and outperforms GPV-1 and VL-T5 across these benchmarks. Our data, code, and web demo are available at <https://prior.allenai.org/projects/gpv2>.

Keywords: General Purpose Vision systems; Webly supervised data

1 Introduction

General Purpose Vision systems (GPVs) [25] are designed to support a wide range of tasks without requiring architectural changes. A task is the application of skills (e.g. localization, captioning) to concepts (e.g. monkey, brown, climbing) in order to map from the input (image, text) to a target output (text, boxes). Given the virtually unlimited number of fine-grained and topical concepts, it is not feasible to provide a GPV with annotations for all skills on all concepts, as even large pre-collected datasets cannot anticipate every need. In this work, we ask: *Can a GPV leverage web image search and skill-concept transfer to massively*

^{*} Equal contribution

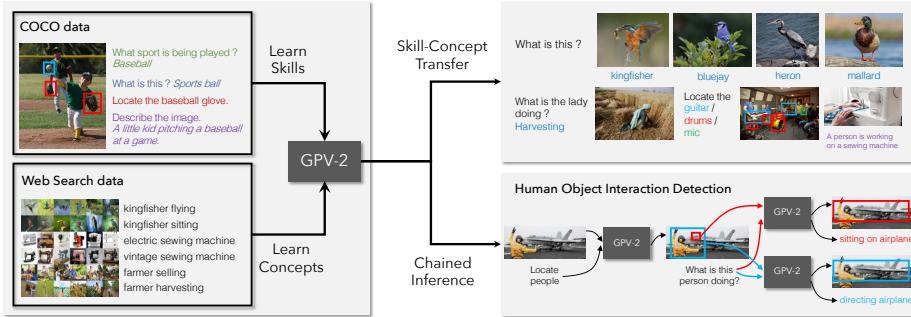


Fig. 1: Learning concepts from the web with GPV-2. We demonstrate webly-supervised concept expansion on two existing GPV architectures (GPV-1 and VL-T5) as well as our proposed GPV-2 architecture. In addition to outperforming previous architectures, GPV-2 expands the inputs to contain bounding boxes which enables support for niche tasks like Human-Object Interaction detection with multi-step inference without any architectural modifications.

and inexpensively expand its concept vocabulary across a variety of tasks? To answer this question, we present a large-scale webly supervised dataset for learning 10k+ concepts, a new benchmark for broader concept evaluation (~ 500) across 5 diverse tasks, and a new GPV architecture that improves cross-task concept transfer and outperforms existing GPVs across multiple benchmarks.

Image search engines provide remarkably good results for millions of queries by leveraging text on the accompanying web pages, visual features from images, and click data from millions of users querying and selecting relevant results each day. They often provide high-quality, decluttered, object- and action-centric images, which can be used to learn powerful visual representations for concepts. Importantly, searches scale easily and inexpensively to thousands of queries. Given the large cost of producing high-quality supervised datasets, scaling today’s manually annotated datasets to support 10,000+ concepts is infeasible for many tasks. In contrast, using Bing search to create WEB10K, a dataset with 1M+ images spanning 10k nouns, 300 verbs, and 150 adjectives with thousands of noun-verb and noun-adj combinations, cost us just over \$150. Moreover, while existing data sources such as ImageNet-22k and YFCC100M are valuable resources, they are static snapshots of a diverse and ever-changing world. For example, these static datasets may not represent specialized categories of interest to a downstream application such as *boysenberry* and will definitely not contain latest concepts such as *Pixel 6* or *COVID-19 home test*. On the other hand, modern web image search engines are designed to serve imagery on-demand and are uniquely positioned to act as a source of training data for novel and latest concepts. While search engine data provides strong supervision for classification, we demonstrate that current GPVs, GPV-1 [25] and VL-T5 [14], are able to learn concepts from web data and improve on other skills as well, such as image captioning. Importantly, we show that even models that already utilize large-scale pretraining corpora such as Conceptual Captions continue to benefit from

using search engine data and can be easily extended to support new concepts relevant in the present day that have little or no coverage in large static corpora.

We also propose GPV-2, a powerful GPV that 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 output boxes and text for any bounding box or for the entire image. These diverse input and output modalities enable GPV-2 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-2 is Language-Based Localization, whereby *all* tasks are based on scoring/ranking/generation using the same text decoder applied to one or more image regions. This ensures that all tasks share the same weights and representations, ranging from the input encoders all the way to the output decoders — resulting in more effective skill-concept transfer for learning from diverse tasks’ datasets. We also propose a re-calibration mechanism to down-weight scores of labels that are disproportionately represented in training, and demonstrate its effectiveness on out-of-domain test datasets for multiple tasks.

Benchmarking the diverse capabilities of large-vocabulary general purpose models is challenging. Most current datasets in computer vision are designed for single tasks. The recently proposed Coco-SCE [25] benchmark is designed to test the skill-concept transfer ability and overall skill competency across five vision skills. However, it is limited to evaluate these competencies on 80 primary COCO concepts. In this work, we present a new benchmark named DCE for broader concept evaluation for the same five skills but now expanding to 492 OPENIMAGES concepts. DCE is an evaluation-only benchmark sourced from OPENIMAGES [42], VISUALGENOME [40] and NoCAPS [2] with new VQA annotations and has been sampled in a way that prevents over-representation of any single category while maximizing representation of infrequent categories.

We evaluate present day GPVs and GPV-2 on three benchmarks: (i) the Coco-sce and COCO benchmarks [25], (ii) the newly presented DCE benchmark; and (iii) the WEB10K dataset consisting of *manually verified* 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-2 outperforms both GPV-1 and VL-T5 across these benchmarks and shows significantly large gains when using web data, particularly for captioning and classification. We also demonstrate how GPV-2 can be chained to perform niche tasks like human-object interaction detection, without any task-specific architecture modifications. Finally, we show how web data can be efficiently used to expand GPV-2’s concept vocabulary to include new visual concepts that are relevant in today’s world such as *COVID-19 vaccination cards* and *N95 masks*, concepts that are infrequent or non-existent in static corpora.

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 that GPVs can learn concepts from WEB10K and transfer this knowledge to other tasks; (c) DCE, a benchmark spanning

5 tasks and approximately 500 concepts to evaluate GPVs; and (d) GPV-2, an architecture that supports box and text modalities in both input and output, improves skill-concept transfer and outperforms existing GPVs. Our code and benchmarks are available at <https://prior.allenai.org/projects/gpv2>, along with a new tool to easily create a web dataset from a list of queries.

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 [51,28] 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 without task-specific components. This approach has become common in natural language processing via text-to-text generative models [64,5,55], and recent work in computer vision has striven towards this kind of generality [17,7,37,49].

Examples of general-purpose computer vision models include VL-T5 [14], which adapts T5 [64] to jointly train on vision+language (V+L) tasks while using a single text-generation head to produce 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-2 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 input and introducing the ability to generate per-image-region text output. 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 V+L tasks such as VQA and captioning. Many other V+L models [83,73,13,47,52] 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 [77] to build recognition systems. As web results improved, works used this data to build object detectors [15,11,46,70,53,82], attribute detectors [21], image taggers [80], large vocabulary categorization models [84,56,24] and fine-grained recognition models [39,57], segmentation models [69,33,72], on-line dataset builders [44], visual reasoning systems [91] 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 GPVs 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, as 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.* [66] build a 9,000 class detector using Imagenet22k classification data [16] by jointly training for the two tasks. Uijlings *et al.* [74] use Multiple Instance Learning to pseudo-label data and train a large vocabulary detector. Recent works build open vocabulary detectors [87,23,32] 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 V+L field has leveraged object detectors as feature inputs [3,89,2], which can be considered as transferring concepts from detection to downstream tasks. Another effective approach is pre-training using image-captions [50,45,47] like Conceptual Captions [67]. 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 V+L tasks [68]. These methods effectively transfer visual knowledge from caption data to tasks like VQA. Recently Whitehead *et al.* [81] 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

Search engines can be leveraged to collect datasets with highly desirable characteristics: (1) **Diversity** — 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*); (2) **Freshness** — Search engines are designed to serve the freshest content on the internet, and often produce very good results for the latest queries (that may not have existed or been popular before; e.g. *COVID-19 vaccination card*, *2022 winter olympics mascot*) which have few/no occurrences in standard vision datasets that tend to be static; and (3) **Concept focus** — 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.

We present WEB10K, a dataset sourced from web image search data with over 10K concepts. WEB10K contains queries with nouns, adjectives and verbs.

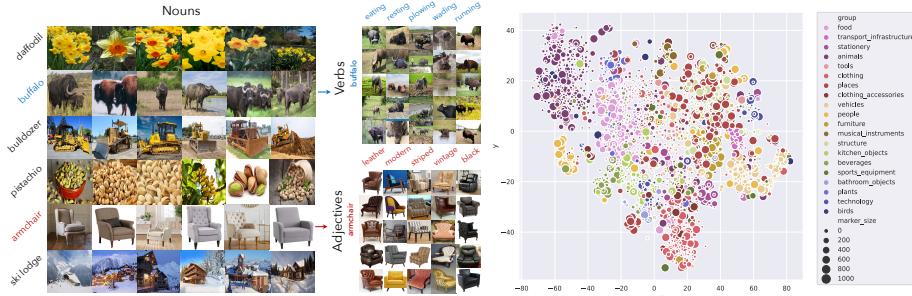


Fig. 2: Concept diversity in WEB10K. **Left:** Besides 10k nouns, WEB10K provides dense coverage of feasible adj-noun and verb-noun combinations to enable learning of fine-grained differences in object appearance due to attributes. **Right:** TSNE [54] plot of Phrase-BERT [78] embeddings of WEB10K nouns 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 concepts which are infrequent/absent in CC.

Nouns. We consider single 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) [67]. We identify candidates using POS tagging and select the most frequent 2,000, and an additional 282 where the second word of the multi-word noun is present in the concreteness dataset (e.g. “street artist”, where “artist” is in concrete nouns). In total, we select 10,211 nouns. Sourcing nouns from a language corpus enables coverage of concepts not commonly covered in vision datasets: over 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 [85], 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 unambiguous and visually recognizable verbs. These verbs improve model performance on action recognition (Supplementary Sec. 8).

Adjectives. We source adjectives from several datasets that have a large number of adjectives [67, 40, 19, 43, 41, 60, 59, 10, 79]. We manually filter out ones that are subjective (“beautiful”), non-visual (“loud”), or relative (“big”). This results in 144 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

Table 1: **Left:** WEB10K statistics (Sec. 3). There are approximately 25 images per concept. **Right:** DCE val and test statistics (Sec. 5).

Type	Count	Subset	Skill	Samples	Images	Categories
Concepts	Nouns: 10211	Val	VQA	5169	2131	295
	Adjectives: 144		Localization	8756	7588	463
	Verbs: 298		Classification	9485	6770	464
	Noun-adjective pairs: 18616		Clss-in-context	9485	6770	464
	Noun-verb pairs: 9243		Captioning	4500	4500	-
Total: 38072 (Nouns + Pairs)		Test	VQA	5281	2160	307
Images	Noun images: 255073		Localization	10586	9986	476
	Noun-adjective images: 465146		Classification	10888	9161	476
	Noun-verb images: 230224		Clss-in-context	10888	9161	476
Total: 950443			Captioning	10600	10600	-
QAs	Templates: 26	Note: Since nocaps [2] annotations are hidden behind an evaluation server, we are unable to provide category counts for captioning.				
	Noun QAs: 1900886					
	Adjective QAs: 930292					
Total: 3291626						

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. See Tab. 1 for detailed statistics.

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 Sec. 3 for all question templates. The 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 crowd-workers to verify that the query is present in the image. We only retain unanimously verified examples (71%) resulting in: Val – 4k images (9k QAs), Test – 8k images (19k QAs). The Train set has about 3M QAs with no manual verification.

4 GPV-2

In this section we present GPV-2, a model combining an object detector with the T5 pre-trained language model. GPV-2 supports additional input and output modalities (and thus tasks) beyond present day GPVs (GPV-1 and VL-T5). It uses the stronger VinVL [89] object detector, uses a shared language decoder (for all tasks including localization) and employs a classification re-calibration approach, which together improve generalization to unseen concepts at test time. **Model design.** GPV-2 takes an image, text, and a bounding box as input. As output, it can produce text for an individual bounding box (the input box, or boxes produced by the visual model) and for the entire image (see Fig. 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

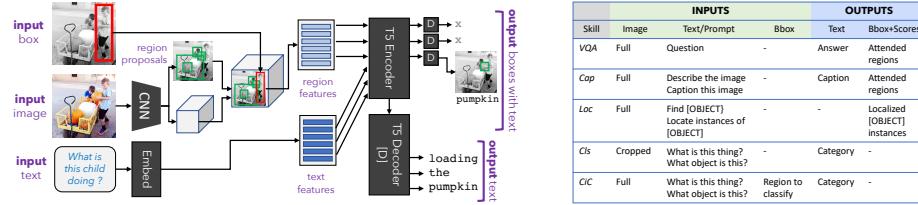


Fig. 3: Left: GPV-2 architecture. **Right:** I/O for 5 skills in Coco and DCE.

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 primarily use the VinVL [89] object detector for our experiments. However the GPV-2 architecture allows us to easily swap in other detectors, and we use features from DETR [7] 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 embedding 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-2. GPV-2'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, 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 a box is scored 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 yield 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, thus *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 [9,8] requires localizing a person, an

object and categorizing their interaction. GPV-2 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 “directing 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. See Supplementary Sec. 2 for an analysis and example.

Pre-training. Recent works have shown that pre-training V+L models on large amounts of data results in large improvements [14,47,89]. We do not have the resources to fully-replicate these setups, but as a partial measure we pre-train GPV-2 for 8 epochs on the CC3M dataset [67], which shows significant gains on our benchmarks. Since GPV-2 is generative, we pre-train it by simply learning to generate the target caption rather than using span masking or other more complex objectives [47,73]. While we use much less data than some V+L works, pre-training on CC3M allows us to verify that GPV-2 still benefits from web data even when exposed to a broad range of concepts during pre-training.

5 DCE Benchmark

The COCO benchmark focuses on 80 object categories and is insufficient for evaluating skills on a wide range of concepts. We introduce the Diverse Concept Evaluation (DCE) benchmark to evaluate GPV models on a large subset of the 600 OPENIMAGES categories across 5 skills: classification (Cls), classification-in-context (CiC), captioning (Cap), localization (Loc), and visual question answering (VQA). See Fig. 3 for the inputs and outputs for each task. We introduce CiC as a natural and unambiguous object classification task (similar to pointing at an object and asking what it is), providing 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. DCE is an evaluation-only benchmark and is not accompanied by a distributionally similar training set.

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

Table 2: Concept expansion with web data. Jointly training on WEB10K + COCO shows consistent gains on DCE and WEB10K benchmarks without adversely affecting COCO performance for 3 different GPVs. GPV-1²⁰ refers to 20 epoch training.

Model	Web data	Coco					DCE					WEB10K			
		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-2	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-2	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

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 frequencies are shown in Supplementary Sec. 4.

Additional VQA annotations. VQA annotations from VG only consist of one answer per question. For each selected VQA sample, we source 9 additional answers from Amazon Mechanical Turk as in standard COCO-based VQA benchmarks [22,4]. Samples where ≥ 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 Sec. 6 for additional experimental 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 [76] on Coco test. DCE uses nocaps [2] for captioning. Due to space constraints, we only report CIDEr-D on the out-of-domain split, as performance on novel concepts is our primary interest. See Supplementary Sec. 11 for results on all splits.

Table 3: Concept scaling using web data: Closed world experiment. To eliminate the effect of VinVL features and CC pretraining, we restrict GPV-2 to Coco-sce trained DETR features. Training jointly with WEB10K still shows massive gains on DCE and WEB10K vs training with only COCO-SCE.

Model	Web data	COCO-SCE												DCE				WEB10K			
		VQA			Cap			Loc			Cls		VQA	Cap	Loc	Cls	All	Noun	Verb	Adj	
		Test	Sn	Unsn	Test	Sn	Unsn	Test	Sn	Unsn	Test	Sn	Unsn								
GPV-2	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	21.1	54.9	13.6	14.0	3.3	11.6	27.1
GPV-2	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	33.4	58.7	26.5	47.0	25.1	43.0	73.0

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-2. **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-2 supports this task.

Training details. We train GPV-2 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 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, resulting in about a third of the data in each epoch being web data. VL-T5 is initialized with a pre-trained checkpoint [14] and GPV-2 is initialized from our checkpoint after CC pre-training. We train GPV-1 to 40 epochs following [25]³.

Concept expansion using web data. Table 2 shows the performance of models 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-2). 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-2 to be the most effective across all three benchmarks. GPV-2 uses less pre-training data and a simpler and cheaper

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

Table 4: Ablating GPV-2. The left-most columns indicate using WEB10K (‘Pre.’ indicates pre-training with WEB10K instead of multi-tasking), CC pre-training, classifier re-calibration (Cb), language-based localization (LBL) (see Sec. 4), and VinVL instead of the DETR detector from GPV-1. The first row shows results for GPV-2, and the lower rows show the differences in scores between ablations and GPV-2. Each component improves performance on DCE.

Web	CC	Cb	LBL	Vin.	Coco				DCE				WEB10K					
					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
Pre.	✓	✓	✓	✓	-0.4	-0.6	0.0	-0.2	0.1	-0.8	-8.2	-1.3	-6.2	-5.5	-31.3	-30.4	-27.6	-35.9
✓	-	✓	✓	✓	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	-11.8	-12.8	0.0	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
✓	✓	✓	✓	-	-8.1	-15.8	6.1	-2.2	-	-9.8	-41.7	-15.0	-17.4	-	-13.8	-13.3	-16.4	-11.7

pre-training strategy than VL-T5. However, it uses more powerful VinVL [89] features and benefits from classifier re-calibration (See Tab. 4). In contrast to VL-T5, GPV-2 can also perform CiC and localization. In contrast to GPV-1, GPV-2 has more shared features and a better pre-trained language model, which help produce large gains across the benchmarks. It also trains much faster than GPV-1 as it can use pre-computed detection features (1 day on 2 GPUs vs. >3 weeks on 4 GPUs). See Supplementary Secs. 10 and 5 for more comparisons and GPV-2 efficiency metrics respectively. GPV-2 also achieves state-of-the-art on the GRIT benchmark [26] at the time of submission (Supplementary Sec. 9).

Closed world evaluation of web data. Table 3 shows results for GPV-2 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 them 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-2 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 pre-train 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-2 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 ablation studies on GPV-2. Table 4 shows results on the validation sets. 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 with-

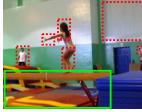
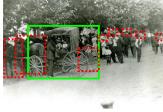
VQA	Captioning	Localization	Classification (cropped)	Classification in Context
What is the type of dress women wearing?  sari scarves	What is happening?  a toddler wearing a hat riding a tricycle. a small child in a hat riding a bike	Find balance beam.  solid lines dotted lines	What is this thing?  gondola motorcycle	What is this?  guacamole broccoli
What color is the burrito?  brown green	Describe this image.  a small blueberry muffin on a yellow plate. a close up of a plate of food on a table	Locate cart in the image.  solid lines dotted lines	What object is this?  harpsichord suitcase	What is this object?  polar bear sheep
Legend: ■ with web ■ without web				

Fig. 4: Qualitative results of GPV-2 on DCE with and without WEB10K: Without web training, GPV-2 can ignore concepts rarely seen in the supervised training data (e.g., ‘balance beam’ top middle) or predict frequently occurring concepts that do not appear in the image (e.g., ‘sheep’ lower right). Web training fixes these issues and allows generalization to rare concepts like ‘sari’ and ‘harpsichord’.

out CC pre-training, and that removing both reduces performance dramatically (>30 points for captioning). This shows that the two approaches are independently effective and complementary at helping models handle new concepts. This is also true to a more modest extent for localization. Using the web data for a second round of pre-training performed better than not using it, but was significantly worse than our multi-tasking framework. Re-calibration is critical for classification, providing a gain of up to 12 points, confirming that models tend to be overly influenced by the concept distribution observed during training. Performance on COCO remains largely unchanged, which is expected as our design choices target performance on unseen concepts. Finally, VinVL significantly out-performs DETR, as expected given its much more extensive training regime.

Human object interaction. To demonstrate the flexibility of GPV-2, we also employ it for human-object interaction detection [8] using the two-stage procedure described in Sec. 4. We fine-tune GPV-2 on the HICO-DET train set for 4 epochs (see Supplementary Sec. 7 for details). GPV-2 gets an AP of 20.6 on the HICO-DET benchmark, which is comparable to a number of other approaches (17.2 [27], 19.8 [75], 20.8 [92], 21.8 [18]). Although recent models [36,93,88] show results up to 32.1 mAP [88], 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-2’s flexibility allows us to get reasonable results by side-stepping complex model design with simple chained inference.

Qualitative results from DCE (Figure 4). Training on WEB10K helps GPV-2 understand rare concepts like ‘sari’ or ‘gondola’, which it is able to use across diverse skills. See Supplementary Sec. 1 for more examples.



Fig. 5: Qualitative results on novel concepts: The predictions of GPV-2 after fine-tuning on COVID-related web data. The model can recognize the new concepts in new images across all skills after training on only ~ 20 images per concept.

Novel concepts case study. A unique advantage of using web-search is the ability to easily and cheaply access new visual concepts that are too specialized or too recent to appear in statically-collected corpora. To demonstrate this we present qualitative results on an experiment to train GPV-2 to learn a number of COVID-19 related concepts. We collect 43 terms related to COVID-19 (e.g., N95 mask, face shield, etc.) and gather a 1000-image train set with a 100-image val set using the same automatic pipeline we used to gather WEB10K. We fine-tune GPV-2 (after it has been trained on COCO and WEB10K) on these examples mixed with a sample of 2000 examples from each COCO train set for 3 epochs.

After fine-tuning, the model achieves 71% accuracy on the new val set compared to only 4% without fine-tuning (performance is initially low since these concepts are too specialized and new to appear in CC, COCO or WEB10K). See some qualitative results in Figure 5 that show that GPV-2 is able to use such recently-introduced concepts when applying multiple skills. Although this is a small-scale qualitative study, it shows that our approach of combining a GPV and web-search data can lead to models that not only understand a wide range of concepts and skills, but can also be efficiently adapted to new visual concepts that become common in the world or that are needed due to the specialized needs of a user. We think this is an exciting avenue for future work in GPVs.

7 Discussion

Extensions. GPV-2 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, through new architectures or training protocols. GPV-2 supports many tasks, but could be extended to handle more modalities (e.g., video) and outputs (e.g., segmentation). Recent work shows promise in this regard [30], potentially enabling transfer of web concepts to a wider range of tasks.

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-datasets and concepts from the web is an efficient and inexpensive option for concept expansion.

Acknowledgements. This work is partially supported by ONR award N00014-21-1-2705.

References

1. Offensive/profane word list. <http://www.cs.cmu.edu/biglou/resources/bad-words.txt> 12
2. Agrawal, H., Desai, K., Wang, Y., Chen, X., Jain, R., Johnson, M., Batra, D., Parikh, D., Lee, S., Anderson, P.: nocaps: novel object captioning at scale. International Conference on Computer Vision pp. 8947–8956 (2019) 3, 5, 7, 9, 10, 12
3. Anderson, P., He, X., Buehler, C., Teney, D., Johnson, M., Gould, S., Zhang, L.: Bottom-up and top-down attention for image captioning and visual question answering. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition pp. 6077–6086 (2018) 5
4. Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Lawrence Zitnick, C., Parikh, D.: VQA: Visual question answering. In: ICCV (2015) 10
5. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krüger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D.M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D.: Language models are few-shot learners. ArXiv **abs/2005.14165** (2020) 4
6. Brysbaert, M., Warriner, A., Kuperman, V.: Concreteness ratings for 40 thousand generally known english word lemmas. Behavior research methods **46** (10 2013). <https://doi.org/10.3758/s13428-013-0403-5> 6
7. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. ECCV **abs/2005.12872** (2020) 4, 8, 12
8. Chao, Y.W., Liu, Y., Liu, X., Zeng, H., Deng, J.: Learning to detect human-object interactions. In: Proceedings of the IEEE Winter Conference on Applications of Computer Vision (2018) 8, 13, 9, 10
9. Chao, Y.W., Wang, Z., He, Y., Wang, J., Deng, J.: HICO: A benchmark for recognizing human-object interactions in images. In: Proceedings of the IEEE International Conference on Computer Vision (2015) 6, 8
10. Chen, H., Gallagher, A.C., Girod, B.: Describing clothing by semantic attributes. In: ECCV (2012) 6
11. Chen, X., Gupta, A.: Webly supervised learning of convolutional networks. In: ICCV (2015) 4
12. Chen, X., Shrivastava, A., Gupta, A.: NEIL: Extracting visual knowledge from web data. In: ICCV (2013) 4
13. Chen, Y.C., Li, L., Yu, L., Kholy, A.E., Ahmed, F., Gan, Z., Cheng, Y., Liu, J.: Uniter: Learning universal image-text representations. ArXiv **abs/1909.11740** (2019) 4
14. Cho, J., Lei, J., Tan, H., Bansal, M.: Unifying vision-and-language tasks via text generation. arXiv preprint arXiv:2102.02779 (2021) 2, 4, 9, 11
15. Divvala, S., Farhadi, A., Guestrin, C.: Learning everything about anything: Webly-supervised visual concept learning. In: CVPR (2014) 4
16. Dong, W., Socher, R., Li-Jia, L., Li, K., Fei-Fei, L.: ImageNet: A large-scale hierarchical image database. In: CVPR (2009) 5
17. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. ICLR **abs/2010.11929** (2021) 4

18. Fang, H., Xie, Y., Shao, D., Lu, C.: Dirv: Dense interaction region voting for end-to-end human-object interaction detection. In: AAAI (2021) [13](#)
19. Farhadi, A., Endres, I., Hoiem, D., Forsyth, D.A.: Describing objects by their attributes. 2009 IEEE Conference on Computer Vision and Pattern Recognition pp. 1778–1785 (2009) [6](#)
20. Fergus, R., Fei-Fei, L., Perona, P., Zisserman, A.: 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) [4](#)
21. Golge, E., Sahin, P.D.: Conceptmap: Mining noisy web data for concept learning. In: ECCV (2014) [4](#)
22. Goyal, Y., Khot, T., Summers-Stay, D., Batra, D., Parikh, D.: Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering. In: CVPR (2017) [9](#), [10](#)
23. Gu, X., Lin, T.Y., Kuo, W., Cui, Y.: Open-vocabulary object detection via vision and language knowledge distillation (2021) [5](#)
24. Guo, S., Huang, W., Zhang, H., Zhuang, C., Dong, D., Scott, M.R., Huang, D.: Curriculunet: Weakly supervised learning from large-scale web images. ArXiv [abs/1808.01097](#) (2018) [4](#)
25. Gupta, T., Kamath, A., Kembhavi, A., Hoiem, D.: Towards general purpose vision systems: An end-to-end task-agnostic vision-language architecture. In: CVPR (2022) [1](#), [2](#), [3](#), [4](#), [8](#), [11](#), [12](#), [9](#)
26. Gupta, T., Marten, R., Kembhavi, A., Hoiem, D.: Grit: General robust image task benchmark. arXiv preprint arXiv:2204.13653 (2022) [12](#), [11](#)
27. Gupta, T., Schwing, A.G., Hoiem, D.: No-frills human-object interaction detection: Factorization, layout encodings, and training techniques. 2019 IEEE/CVF International Conference on Computer Vision (ICCV) pp. 9676–9684 (2019) [13](#)
28. He, K., Gkioxari, G., Dollar, P., Girshick, R.: Mask R-CNN. In: ICCV (2017) [4](#)
29. Hoffman, J., Guadarrama, S., Tzeng, E., Hu, R., Donahue, J., Girshick, R.B., Darrell, T., Saenko, K.: Lsda: Large scale detection through adaptation. In: NIPS (2014) [5](#)
30. Jaegle, A., Borgeaud, S., Alayrac, J.B., Doersch, C., Ionescu, C., Ding, D., Kopputala, S., Brock, A., Shelhamer, E., H’ennaff, O.J., Botvinick, M.M., Zisserman, A., Vinyals, O., Carreira, J.: Perceiver io: A general architecture for structured inputs & outputs. ArXiv [abs/2107.14795](#) (2021) [4](#), [14](#)
31. Jaegle, A., Gimeno, F., Brock, A., Zisserman, A., Vinyals, O., Carreira, J.: Perceiver: General perception with iterative attention. In: ICML (2021) [4](#)
32. Jia, C., Yang, Y., Xia, Y., Chen, Y.T., Parekh, Z., Pham, H., Le, Q.V., Sung, Y.H., Li, Z., Duerig, T.: Scaling up visual and vision-language representation learning with noisy text supervision. In: ICML (2021) [5](#)
33. Jin, B., Segovia, M.V.O., Süstrunk, S.: Webly supervised semantic segmentation. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 1705–1714 (2017) [4](#)
34. Kay, M., Matuszek, C., Munson, S.A.: 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. pp. 3819–3828 (2015) [12](#)
35. Kazemzadeh, S., Ordonez, V., Matten, M., Berg, T.: ReferItGame: Referring to objects in photographs of natural scenes. In: EMNLP (2014) [11](#)
36. Kim, B., Lee, J., Kang, J., Kim, E.S., Kim, H.J.: Hotr: End-to-end human-object interaction detection with transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2021) [13](#)

37. Kim, W., Son, B., Kim, I.: Vilt: Vision-and-language transformer without convolution or region supervision. ArXiv **abs/2102.03334** (2021) [4](#)
38. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014) [8](#)
39. Krause, J., Sapp, B., Howard, A.G., Zhou, H., Toshev, A., Duerig, T., Philbin, J., Fei-Fei, L.: The unreasonable effectiveness of noisy data for fine-grained recognition. ArXiv **abs/1511.06789** (2016) [4](#)
40. Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L.J., Shamma, D.A., Bernstein, M., Fei-Fei, L.: Visual genome: Connecting language and vision using crowdsourced dense image annotations. IJCV **123**, 32–73 (2017) [3, 6](#)
41. Kumar, N., Berg, A.C., Belhumeur, P.N., Nayar, S.K.: Attribute and simile classifiers for face verification. 2009 IEEE 12th International Conference on Computer Vision pp. 365–372 (2009) [6](#)
42. Kuznetsova, A., Rom, H., Alldrin, N., Uijlings, J.R.R., Krasin, I., Pont-Tuset, J., Kamali, S., Popov, S., Malloci, M., Duerig, T., Ferrari, V.: The Open Images Dataset V4: Unified image classification, object detection, and visual relationship detection at scale. arXiv:1811.00982 (2018) [3](#)
43. Lampert, C.H., Nickisch, H., Harmeling, S.: Learning to detect unseen object classes by between-class attribute transfer. 2009 IEEE Conference on Computer Vision and Pattern Recognition pp. 951–958 (2009) [6](#)
44. Li, L.J., Fei-Fei, L.: Optimol: Automatic online picture collection via incremental model learning. International Journal of Computer Vision **88**, 147–168 (2007) [4](#)
45. Li, L.H., Yatskar, M., Yin, D., Hsieh, C., Chang, K.W.: Visualbert: A simple and performant baseline for vision and language. ArXiv **abs/1908.03557** (2019) [5](#)
46. Li, Q., Wu, J., Tu, Z.: Harvesting mid-level visual concepts from large-scale internet images. 2013 IEEE Conference on Computer Vision and Pattern Recognition pp. 851–858 (2013) [4](#)
47. Li, X., Yin, X., Li, C., Hu, X., Zhang, P., Zhang, L., Wang, L., Hu, H., Dong, L., Wei, F., Choi, Y., Gao, J.: Oscar: Object-semantics aligned pre-training for vision-language tasks. In: ECCV (2020) [4, 5, 9](#)
48. Liang, K., Guo, Y., Chang, H., Chen, X.: Visual relationship detection with deep structural ranking. In: AAAI (2018) [6](#)
49. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S.C.F., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. ICCV **abs/2103.14030** (2021) [4](#)
50. Lu, J., Batra, D., Parikh, D., Lee, S.: Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In: NeurIPS (2019) [5](#)
51. Lu, J., Goswami, V., Rohrbach, M., Parikh, D., Lee, S.: 12-in-1: Multi-task vision and language representation learning. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 10434–10443 (2020) [4](#)
52. Lu, J., Goswami, V., Rohrbach, M., Parikh, D., Lee, S.: 12-in-1: Multi-task vision and language representation learning. In: CVPR (2020) [4](#)
53. Luo, A., Li, X., Yang, F., Jiao, Z., Cheng, H.: Webly-supervised learning for salient object detection. Pattern Recognit. **103**, 107308 (2020) [4](#)
54. van der Maaten, L., Hinton, G.E.: Visualizing data using t-sne. Journal of Machine Learning Research **9**, 2579–2605 (2008) [6](#)
55. McCann, B., Keskar, N., Xiong, C., Socher, R.: The natural language decathlon: Multitask learning as question answering. ArXiv **abs/1806.08730** (2018) [4](#)

56. Niu, L., Tang, Q., Veeraraghavan, A., Sabharwal, A.: Learning from noisy web data with category-level supervision. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition pp. 7689–7698 (2018) [4](#)
57. Niu, L., Veeraraghavan, A., Sabharwal, A.: Webly supervised learning meets zero-shot learning: A hybrid approach for fine-grained classification. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition pp. 7171–7180 (2018) [4](#)
58. Otterbacher, J., Bates, J., Clough, P.: 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. pp. 6620–6631 (2017) [12](#)
59. Parikh, D., Grauman, K.: Relative attributes. 2011 International Conference on Computer Vision pp. 503–510 (2011) [6](#)
60. Patterson, G., Hays, J.: Sun attribute database: Discovering, annotating, and recognizing scene attributes. 2012 IEEE Conference on Computer Vision and Pattern Recognition pp. 2751–2758 (2012) [6](#)
61. Pham, K., Kafle, K., Lin, Z., Ding, Z., Cohen, S., Tran, Q., Shrivastava, A.: Learning to predict visual attributes in the wild. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 13018–13028 (June 2021) [10](#)
62. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning transferable visual models from natural language supervision (2021) [5](#)
63. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning transferable visual models from natural language supervision. In: ICML (2021) [5](#)
64. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.* **21**, 140:1–140:67 (2020) [4](#)
65. Ramakrishnan, S., Agrawal, A., Lee, S.: Overcoming language priors in visual question answering with adversarial regularization. arXiv preprint arXiv:1810.03649 (2018) [9](#)
66. Redmon, J., Farhadi, A.: Yolo9000: Better, faster, stronger. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 6517–6525 (2017) [5](#)
67. Sharma, P., Ding, N., Goodman, S., Soricut, R.: Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In: ACL (2018) [5, 6, 9](#)
68. Shen, S., Li, L.H., Tan, H., Bansal, M., Rohrbach, A., Chang, K.W., Yao, Z., Keutzer, K.: How much can clip benefit vision-and-language tasks? ArXiv [abs/2107.06383](#) (2021) [5](#)
69. Shen, T., Lin, G., Shen, C., Reid, I.D.: Bootstrapping the performance of webly supervised semantic segmentation. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition pp. 1363–1371 (2018) [4](#)
70. Shen, Y., Ji, R., Chen, Z., Hong, X., Zheng, F., Liu, J., Xu, M., Tian, Q.: Noise-aware fully webly supervised object detection. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 11323–11332 (2020) [4](#)
71. Suhail, M., Sigal, L.: Mixture-kernel graph attention network for situation recognition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 10363–10372 (2019) [10](#)
72. Sun, G., Wang, W., Dai, J., Gool, L.V.: Mining cross-image semantics for weakly supervised semantic segmentation. In: ECCV (2020) [4](#)
73. Tan, H.H., Bansal, M.: Lxmert: Learning cross-modality encoder representations from transformers. In: EMNLP/IJCNLP (2019) [4, 9](#)

74. Uijlings, J.R.R., Popov, S., Ferrari, V.: Revisiting knowledge transfer for training object class detectors. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition pp. 1101–1110 (2018) [5](#)
75. Ulutan, O., Iftekhar, A.S.M., Manjunath, B.S.: Vsgnet: Spatial attention network for detecting human object interactions using graph convolutions. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 13614–13623 (2020) [13](#)
76. Vedantam, R., Zitnick, C.L., Parikh, D.: Cider: Consensus-based image description evaluation. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 4566–4575 (2015) [10](#)
77. Vijayanarasimhan, S., Grauman, K.: Keywords to visual categories: Multiple-instance learning for weakly supervised object categorization. 2008 IEEE Conference on Computer Vision and Pattern Recognition pp. 1–8 (2008) [4](#)
78. Wang, S., Thompson, L., Iyyer, M.: Phrase-bert: Improved phrase embeddings from bert with an application to corpus exploration. In: EMNLP (2021) [6](#)
79. Wang, S., Joo, J., Wang, Y., Zhu, S.C.: Weakly supervised learning for attribute localization in outdoor scenes. 2013 IEEE Conference on Computer Vision and Pattern Recognition pp. 3111–3118 (2013) [6](#)
80. Wang, X.J., Zhang, L., Li, X., Ma, W.Y.: Annotating images by mining image search results. IEEE Transactions on Pattern Analysis and Machine Intelligence **30**, 1919–1932 (2008) [4](#)
81. Whitehead, S., Wu, H., Ji, H., Feris, R.S., Saenko, K., MIT-IBM, U.: Separating skills and concepts for novel visual question answering. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 5628–5637 (2021) [5](#)
82. Wu, Z., Tao, Q., Lin, G., Cai, J.: 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) pp. 12933–12942 (2020) [4](#)
83. Xu, H., Yan, M., Li, C., Bi, B., Huang, S., Xiao, W., Huang, F.: E2e-vlp: End-to-end vision-language pre-training enhanced by visual learning (2021) [4](#)
84. Yang, J., Feng, L., Chen, W., Yan, X., Zheng, H., Luo, P., Zhang, W.: Webly supervised image classification with self-contained confidence. In: ECCV (2020) [4](#)
85. Yatskar, M., Zettlemoyer, L., Farhadi, A.: Situation recognition: Visual semantic role labeling for image understanding. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 5534–5542 (2016) [6](#)
86. Yatskar, M., Zettlemoyer, L., Farhadi, A.: Situation recognition: Visual semantic role labeling for image understanding. In: Conference on Computer Vision and Pattern Recognition (2016) [10](#)
87. Zareian, A., Rosa, K.D., Hu, D.H., Chang, S.F.: Open-vocabulary object detection using captions. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 14388–14397 (2021) [5](#)
88. Zhang, A., Liao, Y., Liu, S., Lu, M., Wang, Y., Gao, C., Li, X.: Mining the benefits of two-stage and one-stage hoi detection. arXiv preprint arXiv:2108.05077 (2021) [13](#)
89. Zhang, P., Li, X., Hu, X., Yang, J., Zhang, L., Wang, L.J., Choi, Y., Gao, J.: Vinvl: Making visual representations matter in vision-language models. ArXiv **abs/2101.00529** (2021) [5, 7, 8, 9, 12](#)
90. Zhao, J., Wang, T., Yatskar, M., Ordonez, V., Chang, K.W.: Men also like shopping: Reducing gender bias amplification using corpus-level constraints. In:

- EMNLP (2017). <https://doi.org/10.18653/v1/D17-1323>, <https://aclanthology.org/D17-1323> 13
- 91. Zheng, W., Yan, L., Gou, C., Wang, F.: Webly supervised knowledge embedding model for visual reasoning. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 12442–12451 (2020) 4
 - 92. Zhong, X., Ding, C., Qu, X., Tao, D.: Polysemy deciphering network for human-object interaction detection. In: ECCV (2020) 13
 - 93. Zou, C., Wang, B., Hu, Y., Liu, J., Wu, Q., Zhao, Y., Li, B., Zhang, C., Zhang, C., Wei, Y., 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) 13

Webly Supervised Concept Expansion for General Purpose Vision Models

Supplementary Material

Amita Kamath^{*1}, Christopher Clark^{*1}, Tanmay Gupta^{*1}, Eric Kolve¹, Derek Hoiem², and Aniruddha Kembhavi¹

¹ Allen Institute for Artificial Intelligence

² University of Illinois at Urbana-Champaign

The supplementary includes the following sections:

- Sec 1: Qualitative results from GPV-2
- Sec 2: Classification re-calibration analysis
- Sec 3: WEB10K questions and statistics
- Sec 4: DCE sampling details
- Sec 5: GPV-2 efficiency metrics
- Sec 6: Experimental details
- Sec 7: Human Object Interaction experimental details
- Sec 8: Zero-shot verb and attribute recognition
- Sec 9: Performance on the GRIT benchmark
- Sec 10: Comparison between the GPV-2 and GPV-1 architectures when trained on the same data
- Sec 11: Results on all nocaps splits for DCE captioning
- Sec 12: Biases in web data

1 Qualitative results from GPV-2

Qualitative results from GPV-2 are shown in Figure 1. Despite the presence of concepts that are not annotated in COCO (e.g., “Caterpillar”, “Lifejackets”, “Willow”) GPV-2 is able to successfully perform classification, localization, captioning, and visual questioning answering. Visualizations of predictions from GPV-2 on *randomly selected* examples from the Coco, DCE, and WEB10K datasets can be found in additional files in the supplementary materials.

Figure 2 contains an expanded version of Figure 4 from the paper showing the predictions of GPV-2 when trained with and without WEB10K. 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 cups or the background for the class “coffemaker”) while the model trained with WEB10K can localize them accurately.

^{*} Equal contribution

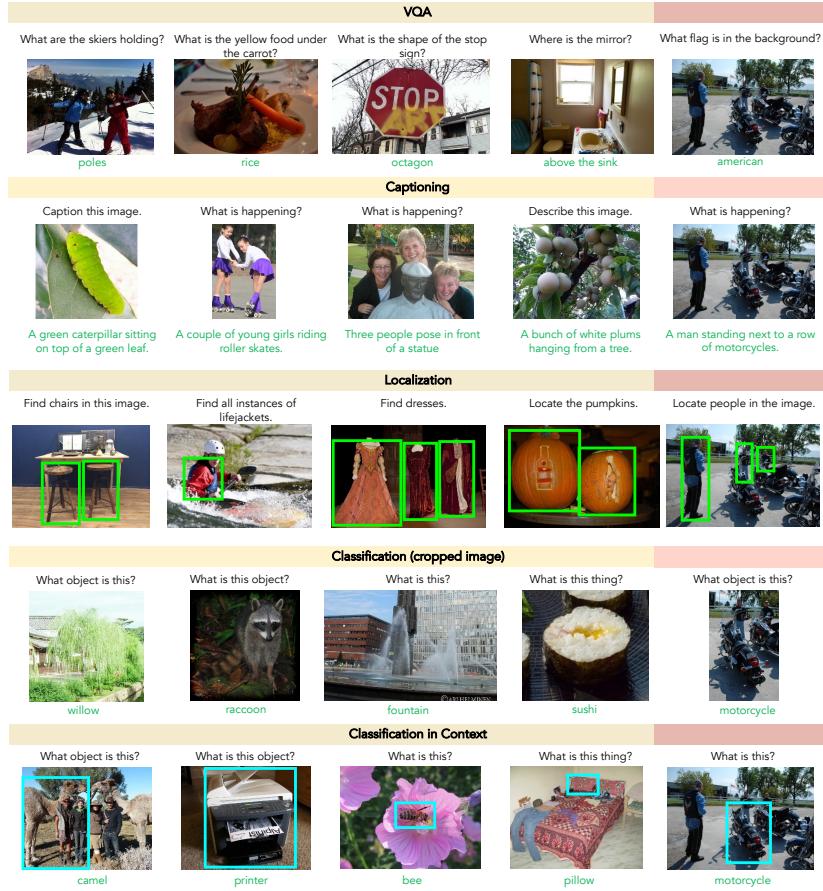


Fig. 1: Qualitative examples for GPV-2. Examples are from DCE val, except for the last image in each row, which comes from Coco val. GPV-2 is able to use concepts that do not appear in the COCO training data across all five skills.

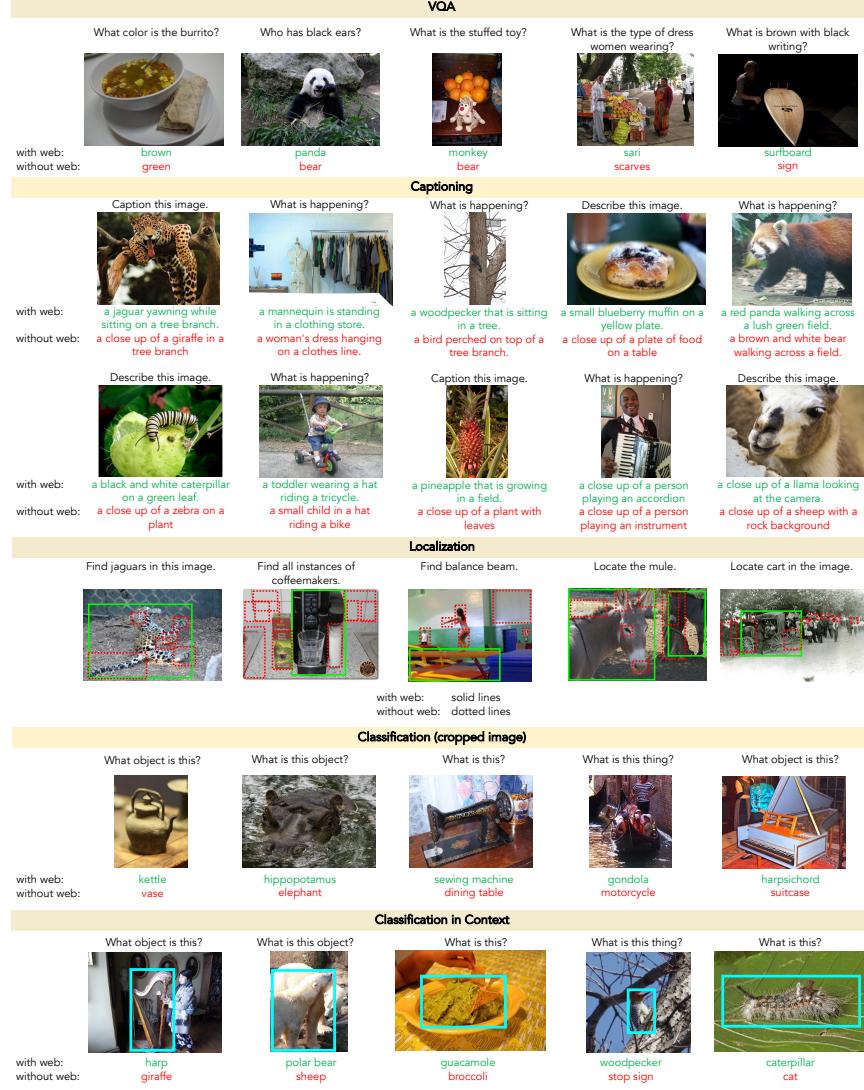


Fig. 2: Qualitative Examples: GPV-2 on DCE, with and without training on WEB10K. The use of WEB10K allows GPV-2 to understand more concepts across all skills, especially for rare concepts such as “red panda” (captioning upper right).

2 Classification re-calibration analysis

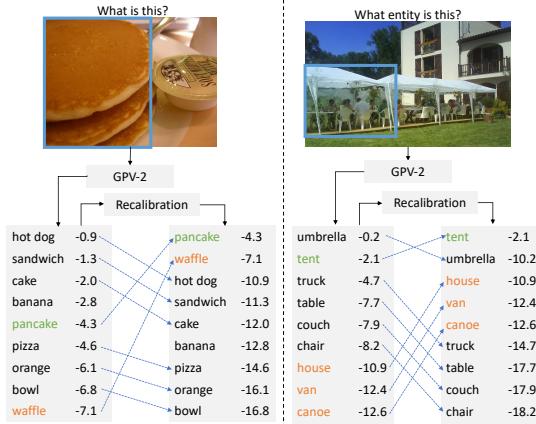


Fig. 3: Qualitative examples of re-calibration. This figure shows two CiC examples, where the left tables show GPV-2’s top 9 predictions and log-probability scores, and the right table shows how the scores and rankings change after re-calibration. The model has a strong preference for answers seen in 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.

In this section, we analyze the classification re-calibration method from Sec. 4. Table 1 shows a breakdown of how GPV-2 behaves on DCE classification with and without re-calibration. Without re-calibration GPV-2 predicts a COCO category for 56% of CiC examples and 65.7% of the CLS examples, even though only 14% of these examples belong to 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.

3 WEB10K questions and statistics

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

Table 1: GPV-2 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.

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

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. To do this, we 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”. Using adjective types is important to because generic questions like “What attributes does this object have?” will have many possible correct answers. Finally, for all query-image pairs, we additionally construct a query whose answer is the entire query. During evaluation, we compute the average accuracy on questions where the 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 possible 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 frequencies 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.

Table 2: Templates for generating web prompts. Templates are grouped by whether they 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.

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. What is DT OBJ doing?
Entire Query	What is this? What is that?

DT → the, this, that

OBJ → entity, object

WH → What, Which

CMD → Describe, State, Specify, Name

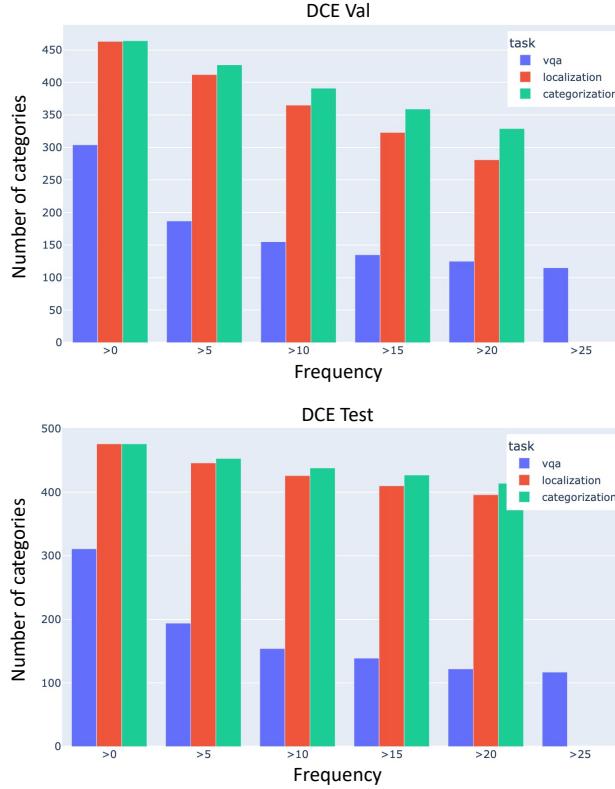


Fig. 4: DCE val and test set category frequencies. Bars at $> x$ indicate the number of categories with at least x samples per category for each DCE skill with publicly available annotations. DCE expands the scope of concept evaluation across skills beyond Coco’s 80 concepts and maximizes representation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

VQA sampling strategy. Co-occurrence of concepts in questions and answers makes the sampling strategy for VQA more nuanced than the one followed for Cls, CiC, and Loc. 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.

Table 3: Number of parameters and FLOPs in GPV-2. Results are shown for both when the image features are pre-computed (top), and when they have to be generated from scratch (bottom).

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

5 GPV-2 efficiency metrics

We report efficiency metrics on GPV-2 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 profiler³ while computing the loss with a single forward pass of 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 [89].

6 Experimental Details

Here we give a more detailed account of how the models are trained. We train GPV-2 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_2 of 0.999, ϵ of 1e-8, and a weight decay of 1e-4. The learning rate linearly warms up from 0 over 10% the training steps and then linearly decreases back to 0. 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

³ https://github.com/facebookresearch/fvcore/blob/main/docs/flop_count.md

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 DCE in which case we use the ranking approach from Sec. 4.

7 Human Object Interaction experimental details

In this section, we provide more details about how GPV-2 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-2 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-2 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].

Finding 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 Zero-shot verb and attribute recognition

Table 4: Learning verbs and attributes from Web10k. We test verb and attribute learning from WEB10K by evaluating GPV-2 without further finetuning on verb (imSitu) and attribute recognition (VAW) benchmarks.

Model	imSitu (top-1 top-5 acc.)			VAW (mAP)		
	Test	Seen	Unsn	Test	Seen	Unsn
GPV-2	10.0 23.0	15.6 33.4	2.5 9.1	53.2	56.9	52.0
GPV-2+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	-	-

In addition to nouns, WEB10K consists of compositions of nouns with verbs and adjectives. To test the learning of verbs and attributes from WEB10K, we evaluate GPV-2 zero-shot on an action recognition dataset (ImSitu actions [86]) and an attribute recognition dataset (VAW [61]), see Table 4. For ImSitu actions we prompt the model with “What are they doing?”. GPV-2 gets 34.7 top-5 accuracy compared to 58.6 from the benchmark authors [86] employing a supervised CNN+CRF approach and 68.6 from a recent supervised model[71] 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. We see no gains on VAW from WEB10K, likely because the model already learns these attributes from VinVL, CC, VQA, and Captioning training data.

9 Performance on the GRIT benchmark

We submit GPV-2 to the Unrestricted track of the GRIT benchmark [26] and achieve state-of-the-art performance at the time of submission. We re-train GPV-2 to include RefCOCO+ [35] in the multi-tasking framework in order to compete on the Referring Expressions Grounding task of the benchmark. See Table 5 for performance results of the model on the test set. The results use the *acc.any.agg.<task>* metric, which averages performance of the model on “same” and “new” source data for each task, as defined in [26]. Note that GPV-2 is trained on more data than GPV-1, and the VinVL backbone used in GPV-2 is trained on OPENIMAGES, which belongs to the GRIT “new” data source (as allowed by the Unrestricted track), contributing to its performance.

The GRIT benchmark website⁴ contains additional information on the data and the models’ ability to generalize to new data sources and concepts, robustness to image distortions, and calibration.

Table 5: Performance on GRIT benchmark, unrestricted test set. GPV-2 competes on four of the seven benchmark tasks: Object Categorization (cat), Object Localization (loc), VQA (vqa) and Referring Expression Grounding (ref). It cannot compete on Segmentation (seg), Person Keypoint Detection (kp), or Surface Normal Estimation (sn). The aggregation takes the average of all seven tasks, assigning 0 to the tasks that models cannot perform. GPV-1 here has not been trained on referring expressions, or with web data.

Model	Detector Backbone	cat	loc	vqa	ref	seg	kp	sn	All
GPV-1	DETR, trained on COCO	33.2	42.7	49.8	26.8	-	-	-	21.8
GPV-2	VinVL, trained on COCO, VG, Objects365 and OpenImages	55.1	53.6	63.2	52.1	-	-	-	32.0

10 Comparison between the GPV-2 and GPV-1 architectures when trained on the same data

We now provide an additional comparison between GPV-2 and GPV-1 in Table 6 using the same training data and detector backbone (frozen DETR), trained only on COCO-SCE. This shows that GPV-2 provides gains over GPV-1 on

⁴ <https://grit-benchmark.org/>

3 tasks purely due to its architecture. In addition, adding web data training to GPV-2 (no other changes) provides further improvements on 2 tasks in-domain. Row [c] corresponds to Table 3 in the main paper.

Table 6: Direct comparison between GPV-2 and GPV-1. Performance on COCO-SCE when trained on the same data and using the same detector backbone.

Model	Web data	VQA	Cap	Loc	Clz
[a] GPV-1	no web	56.4	88.3	63.4	71.5
[b] GPV-2	no web	59.6	88.4	62.2	73.1
[c] GPV-2	with web	59.9	89.2	62.2	73.0

11 Results on all nocaps splits for DCE captioning

See Table 7 for results of the GPVs on all splits of the nocaps dataset [2]: *in-domain*, *near-domain*, *out-of-domain*, and *all*. The out-of-domain results are reported in the main paper, as our focus is on learning novel concepts.

Table 7: Full DCE Captioning results. Training on web data improves performance for all three GPVs, for all splits — even in-domain, which focuses on COCO concepts. GPV-2 achieves the highest performance by a large margin.

Model	Web data	in	near	out	all
[a] GPV-1	no web	69.1	51.4	25.8	49.1
[b] GPV-1 ²⁰	no web	64.4	47.5	23.1	45.3
[c] GPV-1 ²⁰	with web	65.7	51.2	28.6	49.0
[d] VL-T5	no web	70.3	55.9	31.6	53.4
[e] VL-T5	with web	72.0	60.4	45.0	59.1
[f] GPV-2	no web	82.8	79.4	65.4	77.3
[g] GPV-2	with web	85.4	82.6	72.5	81.2

12 Biases in web data

We employ several measures to ensure WEB10K is clean including the “isFamilyFriendly” filter on Bing, removing inappropriate words per a popular black-list [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 [58,34], 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 [90] so we do not recommend using the models in this paper in production settings.