

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

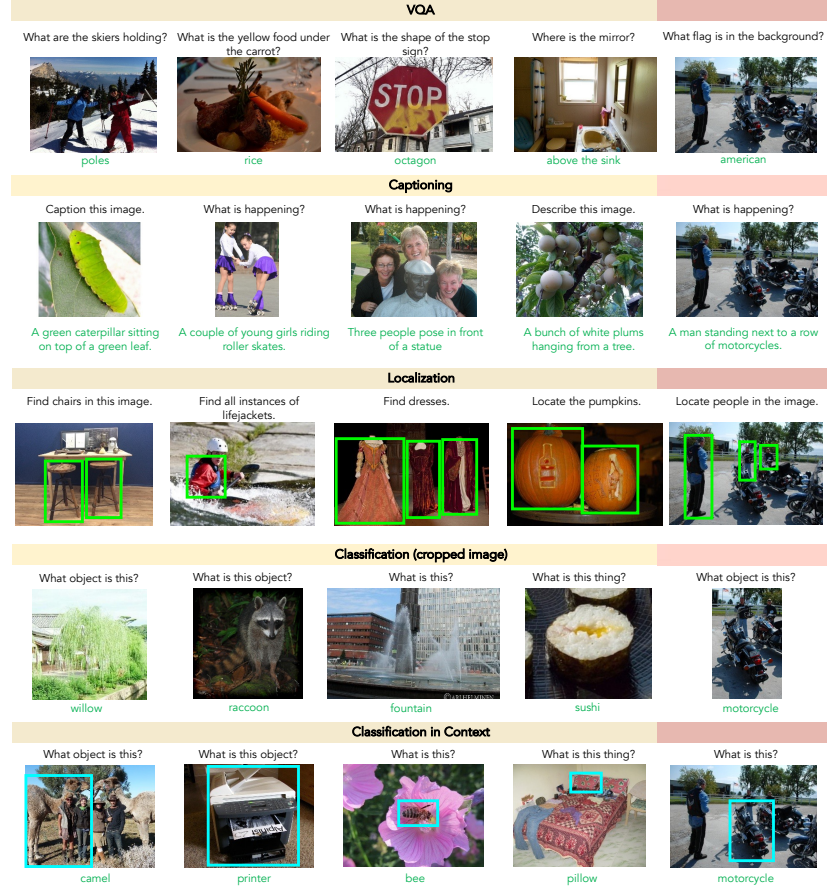


Fig. 1: Qualitative examples for GPV-2. Examples are from DCE val, except for the last image in each row, which comes from COCO val. GPV-2 is able to use concepts that do not appear in the COCO training data across all five skills.


















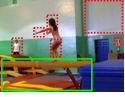
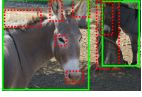






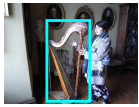




VQA				
What color is the burrito?	Who has black ears?	What is the stuffed toy?	What is the type of dress women wearing?	What is brown with black writing?
				
with web: brown without web: green	with web: panda without web: bear	with web: monkey without web: bear	with web: sari without web: scarves	with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

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

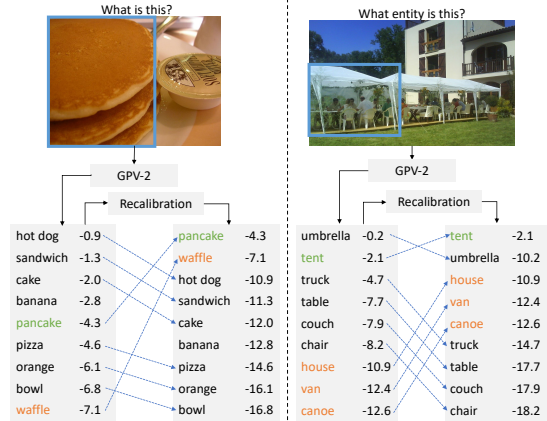


Fig. 3: Qualitative examples of re-calibration. This figure shows two CiC examples, where the left tables show GPV-2’s top 9 predictions and log-probability scores, and the right table shows how the scores and rankings change after re-calibration. The model has a strong preference for answers seen in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

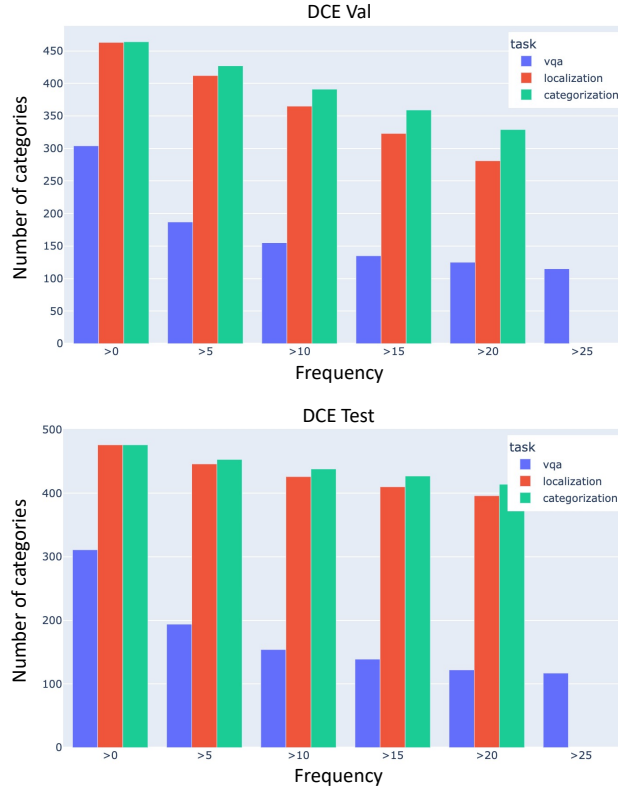


Fig. 4: DCE val and test set category frequencies. Bars at $> x$ indicate the number of categories with at least x samples per category for each DCE skill with publicly available annotations. DCE expands the scope of concept evaluation across skills beyond COCO’s 80 concepts and maximizes representation 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 $3\text{e-}4$, β_1 of 0.9, β_2 of 0.999, ϵ of $1\text{e-}8$, and a weight decay of $1\text{e-}4$. 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)		
	<i>Test</i>	<i>Seen</i>	<i>Unsn</i>	<i>Test</i>	<i>Seen</i>	<i>Unsn</i>	<i>Test</i>	<i>Seen</i>	<i>Unsn</i>
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	<i>VQA</i>	<i>Cap</i>	<i>Loc</i>	<i>Cls</i>
[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	<i>in</i>	<i>near</i>	<i>out</i>	<i>all</i>
[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 blacklist [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.