Towards General Purpose Vision Systems: An End-to-End Task-Agnostic Vision-Language Architecture

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

1. Summary

This supplementary material contains:

- Training details (Sec. 2 and attached **cfg.yaml**)
- Additional ablation results (Sec. 3)
- Task descriptions (Sec. 4)
- Dataset statistics (Sec. 5)
- Randomly sampled qualitative results (Sec. 6 and attached **qualitative_results.pdf**)
- Potential Negative Impact (Sec. 7)
- Code available under Apache-2.0 license at https://github.com/allenai/gpv-1

2. Training Details

Mini-batch sampling strategy. During training, batch sizes are created by randomly drawing samples uniformly across all tasks. Alternatives that require further exploration include drawing samples from only one task in each iteration or ensuring an equal representation of samples from each task in a mini-batch.

Learning from box supervision. For samples that contain a ground truth bounding box (localization task), we compute DETR's Hungarian loss. The Hungarian loss first pairs ground truth and predicted boxes. Hungarian algorithm is used for matching using a linear combination of three cost functions that evaluate for label correctness (relevant or background), high overlap (computed as generalized IoU), and low L1 distance between true and predicted box coordinates. For each pair, the Hungarian loss then minimizes the negative log-likelihood of the ground truth class, generalized IoU loss, and L1 coordinate regression loss. See the attached **cfg.yaml** file for matching cost and loss weights.

Learning from text supervision. For VQA, captioning, and classification tasks which have a text target, we minimize the negative log-likelihood of the ground truth text. This is implemented using cross-entropy with one-hot targets for each word in the text. Ideally, this could be implemented by gathering all samples with text supervision in the mini-batch and minimizing the mean negative log-likelihood (mean computed over samples and not words). However, this led to early overfitting for captioning while other task performances were still improving on the respective validation sets. This is due to a large difference in the number of words in the target text in captioning (upto 20 words) vs other tasks (1-5 words). Therefore we use a weight of 0.05 $(= 1/20)$ for captioning samples while using a weight of 1 for other samples which addressed the early overfitting issue.

Reproducibility. We include model and training hyperparameters in the attached **cfg.yaml** and also our code which will be publicly available under Apache-2.0 license.

Hardware Requirements. We have trained GPV-I models using either $4\times$ Titan RTX (24GB), $4\times$ Quadro RTX 8000 (48GB), or $1\times$ A100 Tensor Core GPU (80GB). On the single A100, GPV-I requires slightly under 60GB of memory but more on the other setups due to multi-GPU training overhead. We have been unable to finetune the full model with 120 batch-size on $8\times$ 12 or 16GB GPUs due to memory issues but it may be possible to further optimize the implementation to do so (e.g using mixed precision training). With batch-size of 120, training GPV-I on all 4 tasks takes 5-7 days depending on the hardware and training split. It may be possible to trade off memory requirements with training time by reducing the batch-size but may require hyper-parameter tuning to retain performance.

Efficiency metrics. GPV-I has 236M parameters, and for a 640x480 image (the image size used during training) on a GeForce RTX 2080 Ti, the inference forward pass yields 289M activations and 139G flops depending on the length of the task description and output text.

3. Additional Ablation Results

In Tab. 1 we show a more detailed task ablation. In addition to performance of GPV-I trained on individual tasks and all tasks, we also provide results for GPV-I trained on VQA and captioning which does not see any concept in $\mathcal{H}_{vqa,cap}$. Hence, we do not expect significant gains on VQA and captioning unseen sets. How-

Model	*Concepts seen during training			VQA			Captioning			Localization			Classification		
	$\mathcal{S}$	$\mathcal{H}_{vqa,cap}$	$\mathcal{H}_{cls,loc}$	Test	*Seen*	*Unseen* ($\mathcal{H}_{vqa,cap}$)	Test	*Seen*	*Unseen* ($\mathcal{H}_{vqa,cap}$)	Test	*Seen*	*Unseen* ($\mathcal{H}_{cls,loc}$)	Test	*Seen*	*Unseen* ($\mathcal{H}_{cls,loc}$)
[a] GPV-I-VQA	✓		✓	55.9	56.5	41.9	-	-	-	-	-	-	-	-	-
[b] GPV-I-Cap	✓		✓	-	-	-	0.855	0.891	0.524	-	-	-	-	-	-
[c] GPV-I-Loc	✓	✓		-	-	-	-	-	-	64.8	69.8	16.4	-	-	-
[d] GPV-I-Cls	✓	✓		-	-	-	-	-	-	-	-	-	**75.3**	**83.1**	0.0
[e] GPV-I-VQA+Cap	✓		✓	57.6	58.3	42.7	0.876	0.913	0.536	-	-	-	-	-	-
[f] GPV-I-Cls+Loc	✓	✓		-	-	-	-	-	-	**64.9**	**70.0**	16.3	74.5	82.2	0.0
[g] GPV-I-VQA+Cap+Loc	✓	✓	✓	**59.6**	**60.2**	**46.6**	**0.911**	**0.949**	**0.559**	**65.1**	**69.3**	**24.1**	-	-	-
[h] GPV-I-Multitask	✓	✓	✓	**58.8**	**59.3**	**47.7**	**0.908**	**0.944**	**0.560**	64.7	68.8	**25.0**	**75.4**	**82.6**	**5.4**

Table 1. **Task-Ablation Results**: We train GPV-I on different combinations of tasks on COCO-SCE split for better understanding of generalization of concepts across tasks. *Concepts seen during training refers to concepts that the model has seen in *any* of the tasks that it was trained on. **Best** and **second best** numbers are highlighted.

	VQA	Cap.	Loc.	Class.
[a] Multitask GPV-I	58.8	0.908	64.7	75.4
[b] *w/o rel. cond.*	59.2	0.926	65.0	75.9

Table 2. **Relevance conditioning ablation.** Counter-intuitively, relevance conditioning slightly hurts the performance. Further exploration is needed to utilize region-relevance scores in text prediction and to learn relevance from text supervision.

ever, GPV-I-VQA+Cap+Loc introduces $\mathcal{H}_{vqa,cap}$ categories during training through the localization task allowing the model to improve the performance on VQA and captioning unseen sets. GPV-I-Multitask results in additional gains by letting the model benefit from classification supervision as well. Similarly, GPV-I-Cls+Loc continues to get zero unseen classification accuracy as $\mathcal{H}_{cls,loc}$ concepts are held out during training but introducing those concepts through VQA and captioning in GPV-I-Multitask leads to improved unseen classification accuracy.

Fig. 1 shows the gains from multi-task training over single task performance for each task and concept-group. Multi-task training improves performance on VQA and Captioning for all groups, improves Classification performance on $\mathcal{H}_{cls,loc}$ without impacting other groups, and improves Localization performance on $\mathcal{H}_{cls,loc}$ at a slight cost to other groups.

We also experimented with removing the relevance-conditioning (Tab. 2) and found the performance to be slightly better without it. This requires further exploration as conceptually we would like relevance scores to be incorporated in text prediction and for text supervision to guide the learning of region-relevance when box supervision is unavailable.

In the vision-language literature, it is common to use task-specific output heads. Tab. 3 compares modality-specific output heads with task-specific heads without changing the rest of the GPV-I architecture. Besides making the architecture more general-purpose, these results show that using modality-specific heads does not sacrifice performance on seen concepts but improves performance on held-out concepts by enabling transfer across tasks.

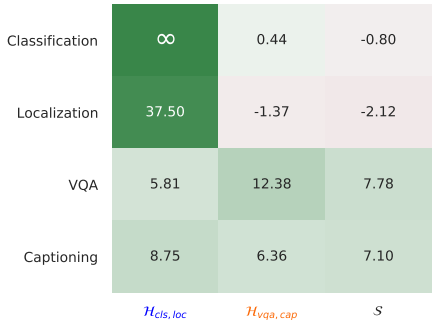

Figure 1. **Impact of Multi-task training:** The heatmap shows the relative gain in average performance of GPV-I-multi-task over GPV-I-single-task for each task and category group. The average performance for any group is computed by averaging performance computed for each category in the group.

4. Task descriptions

Tab. 4 lists the tasks descriptions used to create samples from annotations for each of the 5 tasks. Training on more natural, diverse, and complex task descriptions involving a wide range of skills could lead to improved ability to understand novel descriptions and better zero-shot transfer performance.

5. Dataset statistics

The sizes of COCO and COCO-SCE data splits is shown in Tab. 5. The division of the 80 COCO classes into COCO-SCE splits ($\mathcal{H}_{vqa,cap}$, $\mathcal{H}_{cls,loc}$ and $\mathcal{S}$) is shown in Tab. 6.

6. Randomly sampled qualitative results

The attached **qualitative_results.pdf** contains 10 *randomly sampled* images from the COCO val set, along with COCO-trained GPV-I outputs on all data points associated with each image across all tasks.

Model	Params	VQA			Captioning			Localization			Classification		
		Test	*Seen*	*Unseen*	Test	*Seen*	*Unseen*	Test	*Seen*	*Unseen*	Test	*Seen*	*Unseen*
[a] Head per Task	311M	57.67	58.20	45.86	**0.884**	**0.922**	0.533	62.05	65.76	26.13	74.26	**81.93**	0.00
[b] Head per Modality	236M	**57.73**	**58.22**	**46.91**	0.881	0.915	**0.547**	**62.53**	**66.13**	**27.75**	**74.58**	81.76	**5.10**

Table 3. **Modality-specific vs. task-specific heads.** Comparing GPV-1 (*b*) that uses modality-specific heads to the same architecture but with task-specific heads (*a*). Both models were trained to 20 epochs. Across all tasks, modality-specific heads achieve performance comparable to task-specific heads on seen concepts, while consistently performing better on unseen concepts.

Task	Task descriptions
Captioning	Generate a caption. Generate a description. Describe this image. Describe the image. Caption this image. Caption the image. What is happening in this image? What is happening in the image? What is going on in this image? What is going on in the image? Generate a caption for this image. Generate a caption for the image. Generate a description for this image.
Classification	What is this? What is this object? What object is this? What is this thing?
Localization	Locate [OBJECT]. Locate [OBJECT] in the image. Locate [OBJECT] in this image. Locate instances of [OBJECT]. Locate instances of [OBJECT] in the image. Locate instances of [OBJECT] in this image. Locate all instances of [OBJECT]. Locate all instances of [OBJECT] in the image. Locate all instances of [OBJECT] in this image. Find [OBJECT]. Find [OBJECT] in the image. Find [OBJECT] in this image. Find instances of [OBJECT]. Find instances of [OBJECT] in the image. Find instances of [OBJECT] in this image. Find all instances of [OBJECT]. Find all instances of [OBJECT] in the image. Find all instances of [OBJECT] in this image.
VQA	Questions
RefExp	Referring expressions

Table 4. **Task Descriptions.** For localization prompts [OBJECT] is replaced with the object category name to localize.

		Train	Val
COCO	VQA	443757	214354
	Captioning	414113	202654
	Localization	241035	116592
	Classification	241035	116592

		Train	Val	Test	Seen	Unseen
COCO-SCE	VQA	339411	85858	214354	205138	9216
	Captioning	294028	73773	202654	179402	23252
	Localization	174538	44283	116592	105668	10924
	Classification	174538	44283	116592	105668	10924

Table 5. **Dataset sizes.** The number of examples in each data split of COCO and COCO-SCE for each task. Test combines Test Seen and Test Unseen.

Set	Categories	Set			
$\mathcal{H}_{vqa,cap}$	bed	$\mathcal{S}$	airplane	dog	sandwich
	bench		apple	elephant	scissors
	book		backpack	fire hydrant	sink
	cell phone		baseball glove	fork	skateboard
	horse		bear	frisbee	skis
	remote		bicycle	giraffe	snowboard
	sheep		bird	hair drier	spoon
	suitcase		boat	handbag	sports ball
	surfboard		bowl	kite	stop sign
	wine glass		bus	knife	teddy bear
$\mathcal{H}_{cls,loc}$	banana		cake	microwave	tennis racket
	baseball bat		car	motorcycle	tie
	bottle		carrot	mouse	toaster
	broccoli		cat	orange	toilet
	donut		chair	oven	toothbrush
	hot dog		clock	parking meter	traffic light
	keyboard		couch	person	truck
	laptop		cow	pizza	umbrella
	train		cup	potted plant	vase
	tv		dining table	refrigerator	zebra

Table 6. COCO-SCE **splits.** The 3 disjoint sets that the 80 classes of COCO are split into: $\mathcal{H}_{vqa,cap}$ (held-out from the VQA and captioning tasks in the train/val sets), $\mathcal{H}_{cls,loc}$ (held-out from the classification and localization tasks in the train/val sets) and $\mathcal{S}$ (not held out from any tasks).

7. Potential Negative Impact

A general purpose system aims to solve the same set of tasks with a single architecture that the AI community is creating or has already created separate specialized models for. Hence, any general purpose system inherits the ethical and moral challenges faced by any special purpose system it seeks to replace. For instance, vision-language models are known to reflect [1] and even amplify [3] biases in datasets. In addition, as GPVs become increasingly capable and lower the barrier for training and using deep learning systems, a much larger portion of the population may have access to powerful data-driven capabilities. While increased accessibility may empower many without specialized training in AI or even computer science to benefit from AI tools, this democratization may make regulation of fair and ethical use of such systems challenging. Finally, high computational requirements, which in turn lead to greater energy consumption and carbon emissions [2], need to be kept in check for the health of our planet and its climate.

References

[1] Corentin Dancette, Rémi Cadène, Damien Teney, and Matthieu Cord. Beyond question-based biases: Assessing multimodal shortcut learning in visual question answering. *ICCV*, abs/2104.03149, 2021. 3

[2] Roy Schwartz, Jesse Dodge, Noah Smith, and Oren Etzioni. Green ai. *Communications of the ACM*, 63:54 – 63, 2020. 3

[3] Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. Men also like shopping: Reducing gender bias amplification using corpus-level constraints. In *EMNLP*, 2017. 3