Learning to Localize Objects Improves Spatial Reasoning in Visual-LLMs

Supplementary Material

A. Coordinate Representation Details

We describe our three coordinate representation variants in detail, first focused on bounding-box location format . Consider an image of dimensions (512, 512) containing a cat. Let $(10, 120, 30, 145)$ define the minimal bounding box enclosing the cat in image space ordered as (x1,y1,x2,y2) where (x1,y1) would describe the top left corner and (x2,y2) would describe the bottom right corner of that bounding box. We will use this example in following explanations.

Normalized Floating Point Values would normalize these coordinates using image dimensions to a (0,1) range and directly use normalized values rounded to 4 decimal places. In the given example, the location of the cat would be described $(0.0195, 0.2344, 0.0586, 0.2832)$ which is equal to $(10/512, 120/512, 30/512, 145/512)$ after appropriate rounding.

Integer Valued Binning considers n_b fixed bins across the image that are described by integers 0 to n_b. In our case, for the LocVLM-B version we fix n_b to 224 and for LocVLM-L version we fix n_b to 336. The original bounding-box coordinates are mapped to the range $(0, n_b)$ inspired by prior work [10, 58] using similar binning strategies. In the case of our examples, the location of the cat would be described $(4, 52, 13, 63)$ for $n_b = 224$ which can be easily calculated by remapping the coordinate range as $(n_b \cdot 10/512, n_b \cdot 120/512, n_b \cdot 30/512, n_b \cdot 145/512)$ with integer rounding.

Deviation from Image-Grid based Anchors defines a grid of anchors in image space, selects the anchor closest to the object center, and measures each bounding box coordinate as a deviation from that anchor center. In our case, we set $n_a = 16^2$ for LocVLM-B and and $n_a = 24^2$ for LocVLM-L (motivated by the visual encoder transformer grid size). In both cases, each anchor covers a 14×14 pixel patch. We describe the anchors using (p, q) for $p, q = 0, 1, ...13$. For our example, the bounding box fits the anchor $(0, 4)$ and we represent the bounding box as $(0, 4, 3, 11, 6, 0)$ where the latter four values correspond to pixel deviations from the selected anchor center located at $(7, 63)$ in (224×224) image space.

We also utilize the alternate location form of point values, i.e. (cx, cy) for object center coordinates in image space. Coordinate representations are utilized in the same manner. Instead of four coordinates, we only use two that correspond to the object center. For our given example, the center of the cat would be $(20, 132.5)$ which would be represented similar to the bounding box case.

B. Training Prompt Details

We introduce three instruction fine-tuning objectives that utilize specific hand-crafted templates to generate the target prompts used during training. We discuss in detail, these three objectives presented in Tab. 3 (main paper): LocPred, NegPred, and RevLoc.

For the first two cases, we use a set of 5 templates, one of which is randomly selected for each sample during training.

1. `Where is the object described {category} located in image in terms of {repr}?`
2. `What is the location of object described {category} in terms of {repr}?`
3. `Localize the object described {category} in terms of {repr}?`
4. `Provide a {repr} for the the object described {category}?`
5. `Generate a {repr} for the the object described {category}?`

The placeholder `{category}` is replaced with the relevant ground-truth annotation of each particular object. In the case of COCO dataset, these correspond to one of the 80 COCO categories. For Localize-Instruct-200K (our constructed pseudo-caption dataset), the object pseudo-description is used in place of `{category}`. The `{repr}` can be one of rep_bbox = (x1,y1,x2,y2) bbox or rep_point = (cx,cy) point.

For LocPred, the target is of form ``` ``It is located at {loc}'' ``` while for NegPred, the target is ``` ``There is no such object in the image'' ```. The same five identical prompts are randomly assigned to each objective to ensure no input patterns allow distinguishing between the two targets.

For the case of RevLoc, we similarly sample one prompt from the following set of 3 templates:

1. `Describe the object located at {loc}?`
2. `Provide a caption for object at {loc}?`
3. `What is at location {loc} in image?`

The target is of form ``` ``There is a {category}.'' ``` where category can either be class label or a pseudo-description of that location.

C. Dataset Details

In our work, we first perform blurring of human faces across all our data to preserve privacy in resulting models. These modifications are applied to all our datasets before performing any model training.

As described in Sec. 3.4 (main paper), we explore pseudo-data generation to construct two new datasets, one for object level captions in images and the other for video object labels. We name them first PRefCOCO-100K, and utilize it to construct our Localize-Instruct-200K dataset used for our image level instruction fine-tuning (IFT) objectives. We name the second Pseudo-ActNet and utilize it in our video level IFT objectives.

PRefCOCO-100K uses 95899 images from the COCO dataset and uses an image VQA model (LLaVa [38]) to generate object level descriptions using the COCO object annotations. We first filter images to select those containing unique instances of objects (e.g. only one dog in the image as opposed to multiple dogs). This results in the 95899 images. Next, we ask the VQA model to generate a suitable caption that describes the object category using both its characteristics and relations to surrounding. In detail, we use the exact prompt ``Describe the {category} in this image using one short sentence, referring to its visual features and spatial position relative to other objects in image.'' where category is the ground-truth object label. These obtained object-level captions are used to create question-answer (QA) pairs for the images, resulting in 402,686 such QA pairs.

Following the prompting mechanisms for LocPred and RevLoc described in Appendix B, we generate image-conversation pairs from PRefCOCO-100K, resulting in a human-conversation style dataset we use for training. We refer to this dataset as Localize-Instruct-200K. This contains twice as many image-conversation pairs as the original, given repeated images for both LocPred and RevLoc objectives. This is the main dataset used for our image level training.

For our video domain IFT objective based training, we only use category level labels and leave caption level training as a future direction. We construct Pseudo-ActNet dataset that contains generated bounding-box annotations for all objects belonging to COCO panoptic segmentation dataset [37] categories. Eight uniformly sampled frames are processes per video for annotation. We utilize the pre-trained SEEM [77] model (motivated by [36]) to generate pixel-level panoptic segmentation outputs for each selected frame and convert these segmentations to bounding boxes (panoptic also contains instance level distinction allowing straightforward bounding box extraction). The panoptic outputs (label for each pixel) also allows to obtain an exhaustive list of all COCO dataset categories present in each video - this is necessary to find suitable negative categories for our NegPred objective. Therein, for 8 uniformly sampled frames of each video in the ActivityNet train split, we generate bounding box annotations for all objects belonging to COCO dataset categories and a list of COCO dataset categories not present in those 8 frames. This data is sufficient to implement our IFT objectives on the ActivityNet video dataset with only the videos from the dataset. Our promising results (see Tab. 7) for video-domain IFT using only pseudo-data highlight the data scalability of our proposed framework.

D. Video Architecture & Training

As discussed in Sec. 3.5 (main paper), we introduce two video-domain variants of our framework, LocVLM-Vid-B and LocVLM-Vid-B+. We first detail the architecture common to both variants, followed by specific training procedures.

The overall architecture remains consistent to what is presented in Fig. 2. The visual encoder processes n_f frames independently as images to produce $n_f \times 256$ visual tokens per video (where 256 is tokens generated per image). The spatio-temporal pooling strategy from [42] is utilized to obtain a set of $256 + n_f$ visual tokens per video. In detail, the visual tokens are average pooled across the temporal dimension to obtain 256 spatial tokens and across the spatial dimensions to obtain n_f temporal tokens. These are concatenated to obtain the $256 + n_f$ visual tokens per video. The adaptor layer and LLM remain unchanged - this is straightforward since both these layers perform set-to-set operations independent of input sequence length.

The LocVLM-B-Vid+ variant combines our video level IFT objectives with the training setup from [42]. Given early experiments suggesting insufficiency of fine-tuning only the adapter layer for our IFT objectives, we fine-tune both the LLM and the adaptor layer. We also sample only 8 uniformly spaced frames per video (for compute reasons). The three IFT objectives are modified to suit video domain operation. Given the lack of explicit temporal modelling in our visual backbone and the limited spatio-temporal awareness even within the LLM, we focus on static objects in videos to construct IFT targets. For LocPred and RevLoc, we first filter out objects to select those present only in one of the eight frames or relatively static ones (bounding-box center (x,y) is within a 5 pixel range from their average if present in multiple frames). Then, we obtain the average bounding-box for that object across the frames. These static bounding boxes and negative categories (from the dataset) are used to construct the IFT targets in the same manner as we do for images.

E. Spatial Reasoning Toy Experiment

We present additional details of the toy experiment introduced in Sec. 4.2. We describe the dataset used for evaluation, templates for prompting, and evaluation metric calculation. We also repeat our results from Tab. 4 (main paper) for the left vs right variant here in Tab. 13.

Method	ICL	Acc (All)	Acc (Left)	Acc (Right)
BLIP-2 [33]	✗	45.5	86.1	4.74
LLaVA [38]	✗	55.1	84.5	36.5
Ours	✗	69.5	79.7	59.2
BLIP-2 [33]	✓	14.7	17.8	11.6
LLaVA [38]	✓	55.1	84.7	36.4
Ours	✓	76.5	90.4	61.5

Table 13. **Spatial Reasoning**: We repeat our results for left vs right objects here.

We first construct an evaluation dataset, tagged *COCO-Spatial-27K* containing 26,716 image-question pairs. We build this off the COCO dataset [37] train split through a fully-automated process, utilizing the ground-truth object bounding-box annotations. We first filter out images based on three constraints - this eliminates a large portion of images; hence we elect to use the train split to obtain a considerable quantity of samples after filtering. We first select images containing distinct category object triplets (only one instance occurrence of each object category). For example, an image would contain categories person, dog, and table but only one of each. The second constraint ensures that each object is entirely to the left or right half of the image. This is based on object center not being in the central 20% region. The third constraint is that at least two objects are on opposite sides (i.e. left and right half of image). This provides at least two opposite side object pairs. The ground-truth bounding box annotations enable easy automation of this filtering procedure.

We next discuss our templates for prompting. For two objects on opposite sides tagged `obj_1` and `obj_2`, we use the prompt `Which side of obj_1 is obj_2 located?` and query the model for a response. This is for the direct VQA setting. In the case of in-context learning (ICL) VQA setting, we preprend two examples to the prompt: `Q: Which side of obj_1 is obj_2 located? A: The obj_1 is located to the left of obj_2. Q: Which side of obj_2 is obj_1 located? A: The obj_2 is located to the right of obj_1. Q: Which side of obj_3 is obj_1 located?`. In this case, `obj_3` is the third object, and their ordering is selected such that `obj_1` is on one side, and `obj_2`, `obj_3` are on the opposite side.

Building off standard VQA protocol in [25, 42], we simply query is the terms `left` or `right` are present in the generated outputs, and rate it a success is the target term is present in the generated response. We also visualize some examples for this task in Fig. 3.

F. LLaVA Dataset Analysis

Our results in Tab. 13 indicate unusual disparity in left vs right accuracy numbers, especially in LLaVA [38]. We analyse the training dataset used in this LLaVA baseline to better understand these disparities.

The LLaVA model [38] is instruction fine-tuned on a human conversation style dataset (LLaVA-Instruct-80K). This dataset contains 80,000 image-conversation pairs leading to 221,333 question-answer (QA) pairs across all images (multiple QA for single image). We analyse the presence of keywords related to `left` and `right` concepts that are probed in our spatial-reasoning toy experiment (Sec. 4.2).

We first analyse the exact presence of the words *left* and *right* in the corpus (noting this maybe in different context, e.g. who has the right of way?). Of the 80,000 image-conversation pairs, left and right are present in 1619 (2.02%) and 5001 (6.25%) cases respectively. We provide further statistics of the dataset in Tab. 14 indicating some presence of conversation style training samples encompassing left & right concepts. A large count of the keyword `right` occurs in contexts with different meanings while `left` mostly occurs in its spatial context. We hypothesize that this may be the reason for predicting `left` more often when models are queried with a spatial reasoning related question (i.e. keyword `left` occurs more frequently with *spatial related words* in training corpus).

Template	Left (%)	Right (%)
"the {} "	171 (0.21)	1314 (1.54)
"{} side"	75 (0.093)	110 (0.14)
"to the {}"	80 (0.10)	93 (0.12)

Table 14. We count occurrences of various textual phrases related to left & right concepts in the LLaVA-Instruct-80K dataset.

Therein, we attribute these observed disparities for left vs right accuracy numbers to these artifacts present in datasets used for training underlying LLMs.

G. Limitations & Broader Impact

Our video variant achieves strong performance on VQA tasks but fails to understand temporal locations. In fact, direction use of temporal locations paired with spatial locations results in training collapse for our framework. Extension of our instruction fine-tuning objectives to suitably utilize time coordinates is left as a future direction. In terms of broader impact, while our model uses generic vision and language model architectures, we note that our training data from public datasets may contain biases which should be taken into account when deploying models trained using our framework.

H. Qualitative Evaluation

In this section, we present visual examples showcasing various aspects of our frameworks capabilities. We broadly consider the three distinct settings of spatial reasoning, region description, and generated locations. Note that in all visualizations we blur human faces to make them unidentifiable for privacy reasoings.

Spatial Reasoning: We illustrate examples from our COCO-Spatial-27K dataset highlighting both success cases and failures of our framework. These qualitative results are presented in Fig. 3. In each case, let as tag the two objects within bounding boxes as `obj1` and `obj2`. Following Appendix E, we prompt our framework with each image and `Which side of obj1 is obj2?` and match the response with the ground-truth answer. Correct matches (success cases) on presented on the top row (green) and incorrect matches (failure cases) on bottom row (red). The correct matches indicate the spatial reasoning abilities of our framework across a wide range of image types, including cluttered scenes. The failure cases possibly indicate difficulty at handling truncated / occluded objects.

Region Description: We next illustrate the region description abilities of our model (see Sec. 4.6 for details) in Fig. 4. We query our framework with a set of bounding box coordinate such as `Describe the object located at [22, 114, 86, 154]?` (prompt details in Appendix B) paired with each image. We illustrate the object coordinates as a bounding box (green) in each image. The response of the model presented underneath each image. We highlight invalid responses in red. These qualitative evaluations indicate the ability of our model to not only detect the object present in the queried region, but also describe it in terms of its surrounding: an ability unique to our model in contrast to traditional object classifiers or detectors. At the same time, the generated responses display limitations in terms of object characteristic hallucination and minimal spatial relation (e.g. to the left / right of) based description.

Generated Locations: In our experiments, the tasks of object hallucination and region description directly evaluate the learning resulting from IFT objectives NegPred and RevLoc respectively. In this section, we present some qualitative evaluation to understand the learning resulting from the LocPred objective. These results are visualized in Fig. 5. First, these images present samples from the validation split of COCO modified in a similar manner (i.e. filtering explain in Sec. 4) to our training set for LocPred objective. Each image contains one instance of a particular category. The category is labelled on top of each image, and the ground-truth annotation for the object is in green while the prediction by our framework is in blue. We illustrate the success cases of our model in the top row and failure cases in the bottom row. The success cases indicate strong localization skills across diverse scene involving objects of variable sizes. The failure cases denote difficulty in handling crowded / cluttered scenes and truncated / occluded objects. We also note that direct comparison to classical object detectors is unfair given the down-sampled images (i.e. 224×224 or 336 sized) used by our framework (object detectors use higher resolution images).

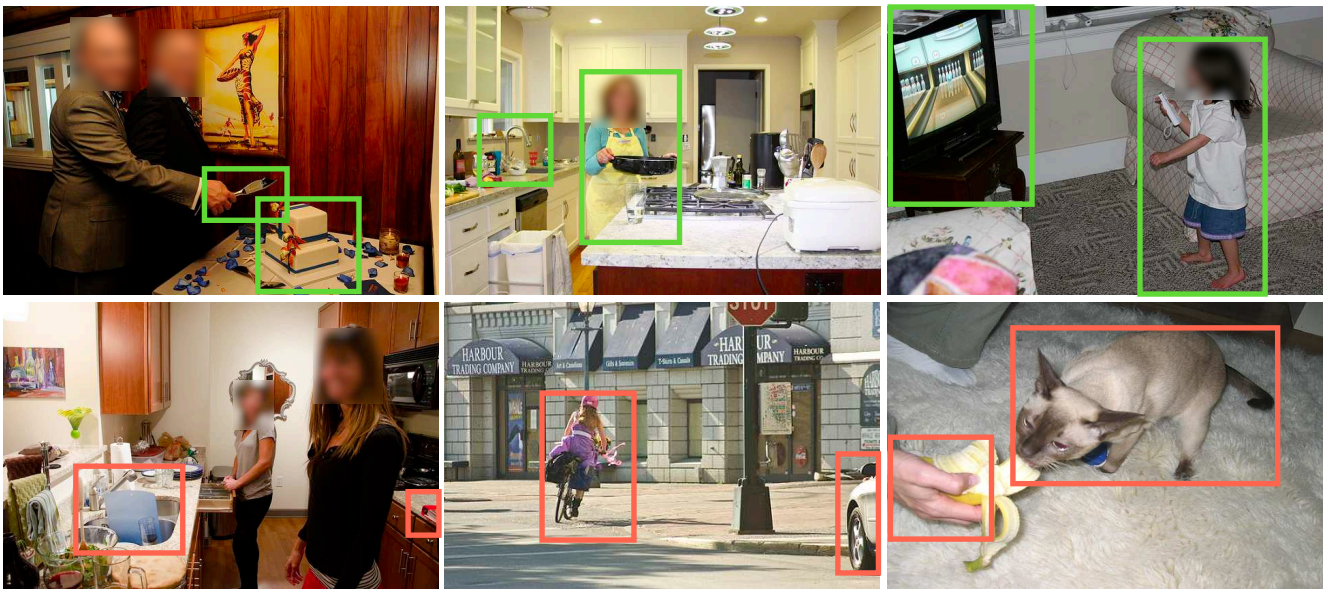

Figure 3. Visualizing Spatial Reasoning: We illustrate example images on which we perform our toy experiment for spatial reasoning (Appendix E). Success cases on top row (green) and failure cases on bottom row (red).

There is a cow that is lying down on a grassy hillside, surrounded by other cows and trees.

There is a cup that is a tall glass, placed on a table next to a pizza.

There is a dog that is a brown and white dog, and it is standing next to a bottle of water, possibly drinking from it.

Figure 4. Visualizing Region Description: Our framework possesses the unique ability of generating representative descriptions for a selected region of an image, input to the model in terms of textual coordinates. We illustrate 3 example images with a bounding box (green) denoting the queried region. The responses generated by our model are underneath each image, with invalid outputs highlighted red.

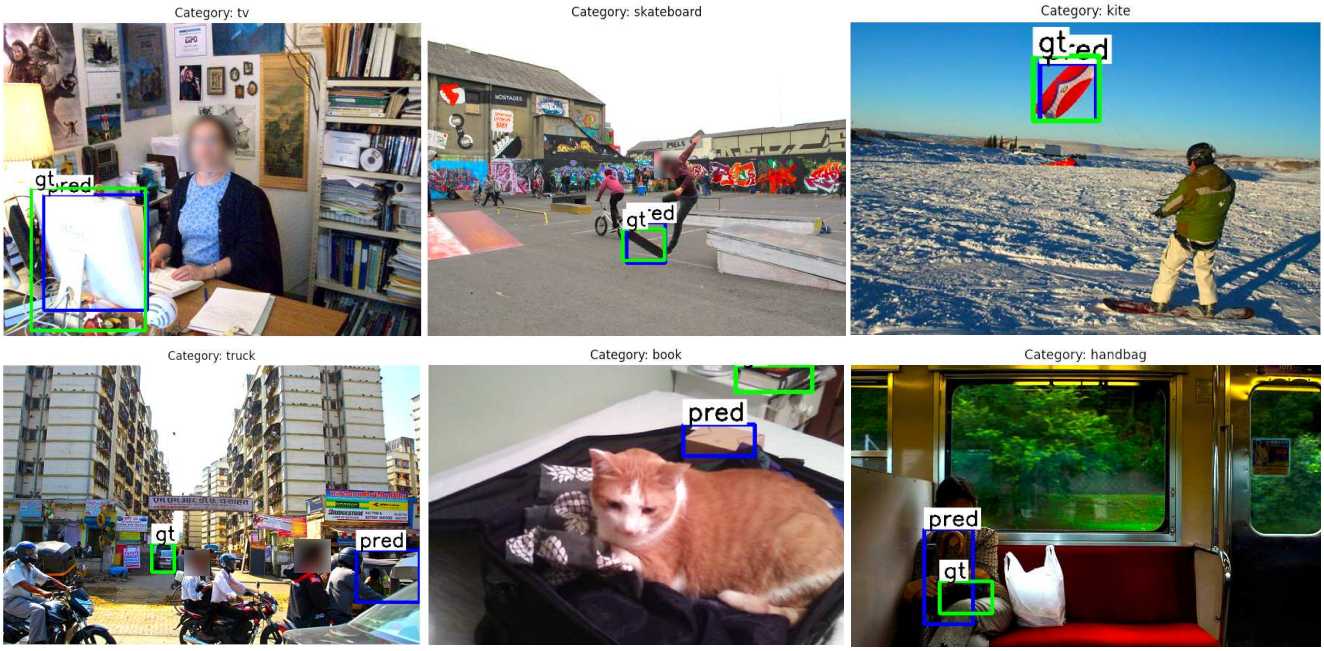

Figure 5. Visualization of LocPred Objective: We illustrate the bounding box locations generated by our framework (blue) when queried with a category label (top of each image) and compare with the ground-truth bounding boxes (green). Success cases on top and failure cases on bottom.