

Learning to Localize Objects Improves Spatial Reasoning in Visual-LLMs

Kanchana Ranasinghe^{1,2} Satya Narayan Shukla¹ Omid Poursaeed¹
 Michael S. Ryoo² Tsung-Yu Lin¹
¹Meta ²Stony Brook University

{kranasinghe, mryoo}@cs.stonybrook.edu, {satyanshukla, opoursaeed, tsungyulin}@meta.com

Abstract

Integration of Large Language Models (LLMs) into visual domain tasks, resulting in visual-LLMs (V-LLMs), has enabled exceptional performance in vision-language tasks, particularly for visual question answering (VQA). However, existing V-LLMs (e.g. BLIP-2, LLaVA) demonstrate weak spatial reasoning and localization awareness. Despite generating highly descriptive and elaborate textual answers, these models fail at simple tasks like distinguishing a left vs right location. In this work, we explore how image-space coordinate based instruction fine-tuning objectives could inject spatial awareness into V-LLMs. We discover optimal coordinate representations, data-efficient instruction fine-tuning objectives, and pseudo-data generation strategies that lead to improved spatial awareness in V-LLMs. Additionally, our resulting model improves VQA across image and video domains, reduces undesired hallucination, and generates better contextual object descriptions. Experiments across 5 vision-language tasks involving 14 different datasets establish the clear performance improvements achieved by our proposed framework.

1. Introduction

Holistic visual understanding requires learning beyond simply content of an image to encompass awareness on spatial locations of objects and their relations [41]. In the context of visual question answering (VQA), such spatial awareness allows better reasoning involving structural and contextual information contained within an image [8].

Since the introduction of powerful large-language models (LLMs) such as GPT-3 [5], Chat-GPT [44], Vicuna [11], and LLaMA [55, 56] that are capable of human style conversation, their visual counterparts such as BLIP-2 [33], LLaVA [38] have enabled novel tasks within the vision modality. However, despite their [33, 38] highly generic visual understanding, these models exhibit poor language-based spatial reasoning [8]. In fact, they fail at simple tasks such as distinguishing whether an object lies to the left or right of another object (see Tab. 4).



Figure 1. We illustrate one unique ability of our model: contextual region description (top). Note the contextual information used in describing the selected region in each image. Explicitly teaching localization to Visual-LLMs also improves spatial awareness in VQA settings (bottom). Color boxes only for illustration purposes.

In the case of contrastive language image models (such as CLIP [46], ALIGN [26]), recent works explore how injecting explicit spatial awareness [39, 43, 48, 74] can enable more holistic visual understanding. In fact, [48] shows how such improved spatial awareness benefits model robustness in adversarial domains. This raises the question of how generative language image models, particularly those connecting LLMs to visual encoders [33, 38] can benefit from such spatial awareness specific training. We refer to models of this category that generate textual outputs given joint image-text inputs (e.g. [33, 38]) as visual-LLMs (V-LLMs).

In this work, we explore location specific instruction fine-tuning objectives that explicitly enforce V-LLMs to

meaningfully process and generate textual image-space coordinates. We hypothesize that such training would lead to improved spatial awareness in these V-LLMs, therein improving performance on VQA tasks. To this end, we propose three instruction fine-tuning objectives that unify location representation with natural language. We also explore optimal representation forms for image-space locations and how pseudo-data generation can be leveraged for efficient scaling of our framework. We name our resulting model as LocVLM.

While the idea of adapting V-LLMs to perform localization related tasks (e.g. detection, segmentation) using V-LLMs has been explored in multiple recent works [30, 45, 58, 66, 68, 73, 75], these approaches depend on task specific architectural modifications or treat localization inputs / outputs differently from natural language. In contrast, our LocVLM focuses on a unified framework treating location and language as a single modality of inputs with the goal of complementing performance in each task. We intuit that processing location represented in textual form would enforce the LLM to select appropriate image regions as opposed to relying on region level features provided by the architecture. At the same time, textual form location outputs promote spatial awareness at language level in a human interpretable manner, in contrast to using secondary heads or specialized tokens for location prediction. Concurrent work in [8] also explores textual location representation with a generic V-LLM architecture similar to our work. Our proposed LocVLM differs with focus on optimal location representation forms, data-efficient pseudo-labelling, and video domain operation.

Our proposed framework exhibits improved spatial awareness in VQA style conversation demonstrated through experimentation on 14 datasets across 5 vision-language tasks: Spatial Reasoning, Image VQA, Video VQA, Object Hallucination, and Region Description. We summarize our key contributions as follows:

- Inject textual spatial coordinate awareness into V-LLMs
- Propose three novel localization based instruction fine-tuning objectives for V-LLMs
- Discover optimal coordinate representation forms
- Pseudo-Data generation for improved region description and scaling to video domain

2. Related Work

Localization in Contrastive Vision Language Models: Foundation vision language models (VLMs) such as CLIP [46] resulted in extensive exploration into language-tied localization in images both under dense (pixel / bounding-box) supervision [16–18, 21, 27, 31, 32, 35, 69, 70] and weak supervision [14, 39, 43, 48, 60, 61, 65, 74, 76]. Recovering explicit localization information within model rep-

Method	Kosmos [45]	Ferret [66]	Shikra [8]	Ours
Unified Arch.	✗	✗	✓	✓
Purely Textual	✗	✗	✓	✓
Pseudo Data	✗	✗	✗	✓
Video Domain	✗	✗	✗	✓

Table 1. Related Work Comparison: A unified architecture, purely textual inputs, pseudo data for scalable learning, and video domain operation distinguishes our work from these prior methods.

resentations has enabled more robust operation for certain tasks [48]. While our work differs from this contrastive setting given our use of LLM based generative predictions, we similarly explore how explicit location information within the language modality can improve V-LLMs.

Visual Large Language Models (V-LLMs): The advent of powerful large language models (LLMs) such as GPT-3 [5], Chat-GPT [44], and PaLM [13], as well as their open-source counterparts BLOOM [52], Vicuna [11], and LLaMA [55, 56], has resulted in direct use of these LLMs for computer vision tasks [22, 53]. Alternate lines of work explore how LLMs can be connected to existing visual foundation models [2, 3, 33, 38, 42, 47], in particular to CLIP visual backbones [46]. While earlier models explored large-scale (millions to billions of samples) image-text training [2, 3], later models [33, 38, 42] scale down on data dependency. LLaVA [38] in particular scales down on pre-training data to under 1 million image-text pairs, and use instruction fine-tuning [59] to enable human-style conversation with visual awareness. This is extended to video domain in [42, 49]. A shortcoming of these models is their lack of spatial awareness or location understanding in image space [8, 12, 20]. Spatial reasoning limitations in generative VLMs are studied in [12, 20]. Similar failures in captioning (and VQA) models are explored in [28]. A solution in [24] proposes code-generation based reasoning. Our work tackles these same limitations but follows an alternate direction of spatial-aware instruction fine-tuning. Another line of recent works [30, 45, 58, 66, 68, 73, 75] tackle this by introducing architectural modifications to explicitly extract region level features that are injected to the LLM as special tokens. While introducing extra tokens and layers, this also separates the localization task from language. In contrast, we use a generic architectures with purely textual location information (i.e. image space coordinates as text). Concurrent work in [8] explores this same idea, but we differ in 3 ways with, a) focus on optimal coordinate representation forms, b) data-efficient pseudo-labelling strategies, and c) video domain operation (see also Tab. 1).

Location Representations: Selecting regions within an image has a rich history in computer vision [40, 57] with greater focus on location outputs since the popularity of object detection [6, 9, 10, 19, 50, 54, 58]. Early anchor-based

methods regress locations from anchor centers [19, 50], followed by direct location regression from object-level features [6, 54]. Recent works explore generative location predictions with diffusion processes [9] and sequence-generation [10, 58]. Ours resembles the latter given our use of an LLM, next token prediction objective, and sequential generation of textual location representations. However, [10, 58] utilize 1000 specialized location tokens (introduced to the LLM vocabulary) corresponding to 1000 bins uniformly spread across image space. While we explore similar binning strategies, in contrast we introduce no additional tokens, focus on purely textual representation of locations, and explore multiple textual location representation forms.

3. Method

Current V-LLMs [33, 38] exhibit weak understanding of spatial locations within images [8]. We explore and benchmark such shortcomings, and propose three novel instruction fine-tuning objectives aimed at overcoming these drawbacks of existing V-LLMs. We build these objectives based on spatial-coordinate based prompting and demonstrate how LLMs can directly both process and generate meaningful numerical coordinates in image-space after suitable training. In the rest of this section we describe our architecture and training framework, followed by coordinate processing & generation, instruction fine-tuning objectives, pseudo-data generation, and video domain operation.

3.1. Architecture and Training

The focus of our work is to explore how spatial localization related training can improve a generic V-LLM such as LLaVA [38]. Therein, our architecture and training framework is inspired from [38]. We use a visual encoder, adapter layer, and LLM stacked sequentially (illustrated in Fig. 2), and follow a multi-stage training strategy similar to [38].

Consider an image $X \in \mathbb{R}^{H,W,C}$ where H, W, C ($= 3$) denote height, width, channels of image and a textual prompt T composed of natural language (asking a question about the image). We define two variants of our model, LocVLM-B and LocVLM-L for better comparison with prior work. We first describe LocVLM-B that processes images with $H = W = 224$. Our visual encoder, ViT-L/14 from CLIP [46], processes the image X to produce a set of 256 visual tokens in \mathbb{R}^{1024} , which are in turn projected to \mathbb{R}^{4096} by an adapter layer (implemented as a linear layer). The LLaMA [55] text tokenizer processes the textual prompt T to produce textual tokens in \mathbb{R}^{4096} . The joint set of visual and textual tokens (of dimension \mathbb{R}^{4096}) are processed by a LLaMA [55] LLM to produce the final set of textual tokens which are in turn untokenized to convert to natural language. The final natural language output is expected to be a suitable response to the input textual prompt, T . In variant LocVLM-L, we use images sized

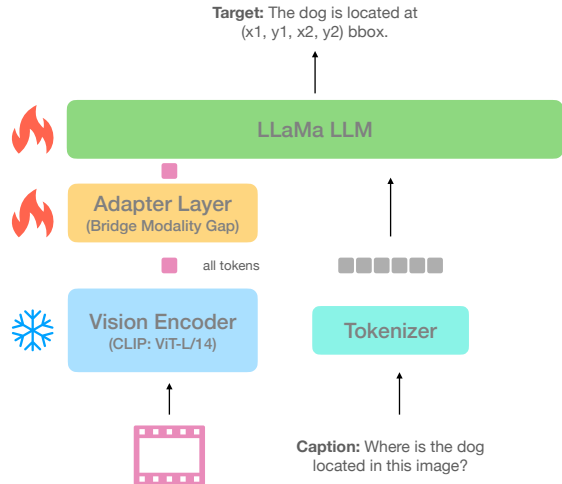


Figure 2. **Architecture:** We present the overall model architecture of our framework which is inspired from LLaVa [38].

$H = W = 336$ resulting in 576 visual token, an adapter layer implemented as an MLP, and the LLM from LLaMA-2 [56]. All other design choices remain unchanged.

We also highlight the BPE tokenization that is employed in our setup. This learned tokenization scheme may split a single word into sub-parts (that alone can appear meaningless to humans) and handles numerical text (including decimal point) as individual tokens (e.g. 12.34 would be split into 5 separate tokens).

In terms of training, we follow a two-stage strategy. Inspired by LLaVA [38], we adopt an initial pre-training stage that only updates weights of the intermediate adapter layer to align the visual encoder outputs with LLM inputs. Next, we jointly instruction fine-tune the adapter layer and LLaMA LLM with our proposed objectives and template-based localization datasets (see Sec. 4.1). Video domain operation introduces an additional phase (see Sec. 3.5).

3.2. Coordinate Processing and Generation

Humans contain the ability to reason about images using image-space coordinates. This is in contrast to existing V-LLMs that can describe the contents of an image elegantly, but lack spatial awareness regarding image contents. We hypothesize that injecting LLMs with additional spatial awareness, through coordinate based reasoning could improve their generic reasoning ability as well. To this end, we introduce our first goal of directly using textual coordinate based image locations in both natural language prompts and LLM generated outputs. For textual coordinates, we explore three different representation forms:

1. Normalized Floating Point Values
2. Integer Valued Binning (across image dimensions)
3. Deviation from Image-Grid based Anchors

CR	GQA (Acc)	RD (METEOR)	A-QA (Acc)
NFP	46.1	19.6	37.1
IVB	47.3	20.7	37.4
DIGA	47.0	20.8	37.3

Table 2. Ablation on Coordinate Representation (CR) methods: we compare each of the three proposed CR variants, namely normalized floating point values (NFP), integer valued binning (IVB), and deviation from image-grid based anchors (DIGA).

For image locations, we explore point based (e.g. center coordinates $[cx, cy]$ of object) and bounding box based (e.g. top-left and bottom-right extreme coordinates of object region $[x1, y1, x2, y2]$) forms. We next discuss the three representations for coordinates used for either location.

Normalized Floating Point Values calculates absolute image coordinates and normalizes with image dimensions to a $(0, 1)$ range. We use a 4 decimal point representation for these floating point values. While this representation is simple and generic, given the nature of BPE tokenization, each individual coordinate will be represented by up to 6 tokens.

Integer Valued Binning discretizes the absolute image coordinates to one of n_b ($=224, 336$ for variant B & L respectively) bins spread uniformly across the two image dimensions. Based on the binning parameter, n_b , each coordinate will be represented some number of tokens, in our case up to 3 (less than the floating point variant).

Deviation from Image-Grid based Anchors is motivated from prior object detection works that estimate an initial anchor followed by deviation from that anchor center to estimate bounding box coordinates. We follow a similar setup, where one of n_a anchors is predicted by the model, followed by deviation of coordinate from that anchor center. Our intuition is that, given the sequential next-token prediction setup of LLMs, such a two-stage strategy would lead to faster learning and more accurate coordinates.

We refer to Appendix A for further details on each variant. In Tab. 2, we ablate each representation format on three different tasks (see Sec. 4.7 for more details) of image VQA (GQA), region description (RD), and video VQA (A-QA). Our experiments indicate optimal performance for integer valued binning (IVB). In all following experimentation, we fix our coordinate representation to IVB.

3.3. Instruction Fine-Tuning Objectives

Given suitable coordinate representations, we now have a mechanism to directly prompt LLMs with image locations in textual form. Our second goal is to build training objectives using these image coordinates that directly inject spatial awareness into V-LLMs. We propose three instruction fine-tuning objectives for this purpose.

Let us first revisit the visual instruction fine-tuning methodology in [38]. Building off the COCO dataset,

Objective	Prompt	Target
LocPred	Where is obj1?	It’s at $(x1,y1,x2,y2)$.
NegPred	Where is obj2?	There’s no obj2.
RevLoc	Describe (cx,cy)	<i>Detailed description</i>

Table 3. We summarize our three distinct instruction fine-tuning objectives. Refer to appendix (Appendix B) for exact natural language prompts and targets used for training. For illustration, we use both point and bounding-box based image locations here.

they construct a VQA dataset containing conversation style question-answer pairs relevant to each COCO image. Question-answer pairs are generated using an LLM that is fed with the ground-truth bounding-box annotations for each image. Inspired by this setup, we build a similar spatial-VQA dataset using images and annotations of the COCO dataset, but instead of LLM prompting, we utilize hand-crafted templates and pseudo-captions (discussed in Sec. 3.4) to generate conversations.

We propose three types of question-answer pairs that relate to our three instruction fine-tuning objectives: Location Prediction (LocPred), Negative Prediction (NegPred), and Reverse-Location Prediction (RevLoc). See Tab. 3 for examples. Considering the LLM based final text generation in our architecture, we utilize next-token prediction loss to achieve each objective during our training.

Location Prediction: Given an object category, we query the model to generate a point or bounding box localizing that object in the image. The object category and bounding box are derived from the COCO train set annotations. To avoid object mismatches (i.e. multiple object of same category), we first filter images containing only a single object of a given class.

Negative Prediction: Using the same prompt templates as in *LocPred* above, we query the model to generate a point or bounding box localizing a specified object in the image. However, in this case we select an object category not present in the image and accordingly provide a target text of “no such object in image”. For each image, we utilize COCO bounding-box annotations to discover objects (belonging to COCO classes) that are not present in that image.

Reverse-Location Prediction: We perform the reverse of *LocPred* here. Given a point or bounding box in image space, we query the model to describe the object in that location. The bounding box and object category are derived from the COCO train set annotations.

While introducing three novel train objectives aimed at injecting location information, we highlight that our proposed framework relies on training data (i.e. human annotations) identical to those used by LLaVA [38]. We do not use any additional ground-truth annotations for training. Next we explore how we could augment the generality of our framework while limiting to this same annotated data.

3.4. Pseudo-Data Generation

We introduced three train objectives, each utilizing template based conversations as prompts and targets. However, our reliance on categories of COCO dataset limits the object vocabulary seen during training. Therein, we propose a pre-trained V-LLM based pseudo-data generation strategy. In fact, we utilize our model after stage one training as the V-LLM leading to a form of self-training based learning. Given the abundance of only image-level annotated datasets (i.e. no bounding box ground-truth), we also explore how an object-detector generated pseudo-bounding boxes could augment our framework.

Self-Training: Given an image and bounding box annotations from the COCO train set, we prompt the V-LLM to caption each distinct object in the image. In order to prevent ambiguous object queries, we filter images to select only those containing at most one instance of a single category. We additionally prompt the V-LLM to describe the object using relational information (i.e. relative to other objects in image). This process provides us a dataset with object level bounding boxes and descriptive captions that are not limited to the COCO object categories (dataset details in Appendix C). In turn, we use this data generated by the V-LLM (our stage one model) to further improve performance of our framework. We modify each of our three train objectives (in Sec. 3.3) to utilize these image-specific pseudo-captions instead of the generic dataset level category labels.

Weak Supervision: We explore how datasets containing no object level annotation (e.g. video classification / VQA datasets) could be leveraged to adapt our framework into domains beyond images. Therein, we utilize an off-the-shelf panoptic segmentation framework from SEEM [77] to generate pseudo-bounding boxes for selected object categories within any image as well as exhaustive pixel level labels (enabling negative class identification). We leverage this setup to extend our introduced train objectives to the video domain as well.

3.5. Video Domain Operation

Inspired by the simple modifications to LLaVA [38] in [42] enabling video domain operation, we follow a similar strategy of modifying our LocVLM-B architecture to process videos while introducing no additional components. The visual backbone process multiple video frames individually (as images) and resulting tokens are averaged across spatial (S) and temporal (T) axes to obtain $S + T$ tokens. These are processed by the adapter layer and LLM to generate the textual outputs. Further details on our video architecture are discussed in Appendix D.

In addition to our two training phases discussed in Sec. 3.1, we introduce a third video instruction fine-tuning stage using a dataset we derive from ActivityNet [23]. Fol-

lowing [42], only the adapter layer is fine-tuned leaving all other parameters frozen. This resulting model is referred to as LocVLM-Vid-B.

We next introduce video variants of our three instruction fine-tuning objectives focused on static objects in videos. We utilize our proposed pseudo-labeling strategy to generate necessary video annotations and train both the adapter layer and LLM to obtain a second video model tagged LocVLM-Vid-B+. Further details on our video fine-tuning objectives are presented in Appendix D.

4. Experiments

In this section, we present experimental results to highlight existing weaknesses of SOTA V-LLMs and how our proposed framework addresses these issues. We also evaluate on standard VQA benchmarks across domains to showcase the better reasoning abilities of our model and highlight novel abilities of our framework.

4.1. Experimental Setup

Datasets: We utilize the COCO dataset [37] and our model (post stage one training) to construct a localization related VQA dataset as outlined in Sections 3.3 and 3.4. We name this dataset *Localize-Instruct-200K*. In detail, this contains LocPred and RevLoc question-answer pairs that use pseudo-captions instead of COCO categories as well as NegPred. We define a second video dataset, *Localize-ActivityNet*, containing question-answer pairs constructed from Activity-Net pseudo-bounding boxes following Section 3.5. Our models are primarily trained on our Localize-Instruct-200K dataset. Our stage one training uses CC3M dataset [7]. Additionally, our Localize-ActivityNet dataset and ActivityNet dataset [23] are used for video domain training.

Training: We train our models on 8xA100 GPUs (each 80GB) following a two-phase training schedule. Our first phase trains on CC3M [7] following the setup in [38]. The second phase uses our Localize-Instruct-200K dataset and trains for 10 epochs with a batch size of 64, ADAM-W optimizer with initial learning rate $2e - 5$, 0.3 warm-up ratio, and cosine-decay learning rate schedule. Both the training phases we conduct use standard next-token-prediction loss used in LLM training.

Evaluation: During evaluation, following standard protocol [38], we iteratively generate next tokens, given visual and textual inputs. The LLM output is a distribution across the entire token vocabulary. The next token is selected through multinomial sampling of this output using a softmax temperature term of 0.2 during normalization.

4.2. Spatial Reasoning: A Toy Experiment

We investigate spatial reasoning abilities of two SOTA V-LLMs, LLaVA [38] and BLIP-2 [33], using a simple toy experiment. We create an evaluation dataset from COCO annotations containing images with distinct category object triplets (only one instance occurrence of each object category), where each object is entirely to the left or right half of the image and two objects are on opposite sides. The ground-truth bounding box annotation are utilized to automate this dataset creation procedure. This evaluation set, referred as *COCO-Spatial-27K*, contains 26,716 image-question pairs (see Appendix C for details). We introduce two evaluation settings, direct VQA and in-context learning (ICL) VQA to understand spatial reasoning abilities of these models. In direct VQA, given an image we query the model whether an object lies above or below another object. In ICL VQA, before a similar final query, we provide two example question-answer pairs (involving the other two objects in the image) in the same format as our query. Refer to Appendix E for further details on task. We perform the same for objects in top vs bottom halves of images.

These results are presented in Table 4. Our results indicate near random performance for existing V-LLMs. For the case of LLaVA, we perform keyword (*left* and *right*) frequency analysis on its instruction tuning dataset (LLaVA-Instruct-80K dataset) to verify the presence of terms *left* and *right* in its training corpus. These keywords are present in 0.37% and 1.13% of its conversations respectively (see Appendix F for more) indicating presence of these concepts in the image-text training corpus. In contrast to these methods, our proposed framework notably improves performance over both BLIP-2 [33] and the LLaVA baseline [38].

4.3. Image VQA

Image VQA involves correctly answering natural language questions regarding content within an image. We evaluate our model for Image VQA on two standard datasets, GQA and VQAv2. The GQA dataset focuses on questions requiring compositional reasoning, particularly involving surrounding information of objects within an image. We evaluate on its test-dev split containing 12,578 image-question pairs. The VQAv2 dataset contains open-ended questions about each image that require an understanding of vision, language and commonsense knowledge to answer. We use its validation split containing 214,354 image-question pairs for our evaluation. For each dataset, we follow standard V-LLM evaluation protocol following [33, 42] and report top-1 accuracy metric. Our results in Tab. 5 indicate clear improvements for LocVLM over our baseline and prior work, establishing the usefulness of our proposed framework. The closest to our work, Shikra [8] achieves performance competitive to our LocVLM-B, but

Method	ICL	All	Left	Right	All	Above	Below
BLIP-2 [33]	✗	45.5	86.1	4.74	49.2	50.4	48.6
LLava [38]	✗	55.1	84.5	36.5	58.9	57.8	59.3
Ours	✗	69.5	79.7	59.2	65.4	64.2	65.9
BLIP-2 [33]	✓	14.7	17.8	11.6	15.8	16.5	15.2
LLaVa [38]	✓	55.1	84.7	36.4	58.2	57.7	58.5
Ours	✓	76.5	90.4	61.5	74.1	73.5	74.4

Table 4. **Spatial Reasoning:** We report accuracy (%) on a spatial localization dataset derived from COCO annotations to highlight weak spatial awareness of existing V-LLMs. We query these models to answer whether one object is to the left or right / above or below of another object. The SOTA V-LLMs evaluated exhibit close to random performance. Our proposed setup outperforms existing methods. Ours refers to LocVLM-B variant.

Method	LLM	VS	Zero-Shot	GQA	VQA-V	VQA-T
SR [4]	-	-	✗	62.1	72.9	-
Shikra [8]	7B	224	✗	-	75.3	77.4
LLaVA-v1.5	7B	336	✗	62.0	78.1	78.4
LocVLM-L	7B	336	✗	63.5	78.2	78.6
LLaVA-v1	7B	224	✓	44.7	49.8	49.3
LocVLM-B	7B	224	✓	47.3	50.3	50.8
Viper-GPT	175B	-	✓	48.1	-	-
BLIP-2	11B	-	✓	44.7	54.3	53.9
LLaVA-v1.5	7B	336	✓	48.7	55.7	55.3
LocVLM-L	7B	336	✓	50.2	55.9	56.2

Table 5. **Image VQA Results:** We report accuracy (%) on the test-dev split of GQA dataset (GQA) and the validation / test splits of VQAv2 dataset (VQA-V / VQA-T). Our proposed LocVLM improves over prior works achieving state-of-the-art performance.

unlike ours they use VQA datasets (containing similar domain question-answer pairs) during training.

4.4. Video VQA

Our model is also applicable to video tasks following our video domain adaptation described in Sec. 3.5. We simply adopt the additional video instruction fine-tuning phase from [42] on the ActivityNet dataset after our initial two phases of training to obtain LocVLM-Vid-B. This third phase involves fine-tuning only the adapter layer of our model. We also explore video variants of our IFT objectives that train both adapter layer and LLM. The resulting model is termed LocVLM-Vid-B+.

Video VQA focuses on correctly answering questions regarding a given video that require spatio-temporal awareness to answer. We evaluate our video-adapted model on the task of zero-shot video VQA on four benchmark datasets, ActivityNet-QA, MSRVT-QA, MSVD-QA, and TGIF-QA. We evaluate on the validation splits of these four datasets. ActivityNet-QA videos cover a wide range of complex human activities relevant to daily living with its question-answer pairs focusing on long-term spatio-temporal reasoning. MSRVT-QA builds off the MSRVT dataset that contains web videos covering a comprehensive

Method	Zero-Shot	ActivityNet-QA	MSRVTT-QA	MSVD-QA	TGIF-QA
JustAsk [63]	✗	38.9	41.8	47.5	-
FrozenBiLM [64]	✗	43.2	47.0	54.8	-
VideoCoCa [62]	✗	56.1	46.3	56.9	-
Flamingo [2]	✓	-	17.4	35.6	-
BLIP-2 [33]	✓	-	17.4	34.4	-
InstructBLIP [15]	✓	-	25.6	44.3	-
FrozenBiLM [64]	✓	24.7	16.8	32.2	41.0
Video Chat [34]	✓	26.5	45.0	56.3	34.4
LLaMA Adapter [72]	✓	34.2	43.8	54.9	-
Video LLaMA [71]	✓	12.4	29.6	51.6	-
Video-ChatGPT [42]	✓	35.2	49.3	64.9	51.4
LocVLM-Vid-B	✓	37.4	51.2	66.1	51.8

Table 6. **Video VQA Results:** Our proposed LocVLM-Vid-B improves over Video-ChatGPT [42] and achieves state-of-the-art results (Top-1 Accuracy %) across four different video VQA benchmarks. Note the zero-shot setting of all these evaluations.

range of categories and diverse visual content. MSVD-QA is a similar dataset building off the MSVD dataset. TGIF-QA contains question-answer pairs from a dataset constructed of animated GIFs. For each dataset, we report the accuracy metric following evaluation protocol in [42]. Our results on these four datasets reported in Tab. 6 demonstrate state-of-the-art performance of our proposed LocVLM-Vid-B, with consistent improvements over the baseline from [42]. Here we use the LocVLM-Vid-B variant for fairer comparison with the baseline from [42]. We attribute the performance gains exhibited by our model to its stronger spatial awareness (see Sec. 4.2). Particularly in the case of video understanding, awareness of content at spatial level of each frame is significant to understand object motions and interactions [1, 51]. We also report more results involving additional model variants in Tab. 7.

4.5. Object Hallucination

Current state-of-the-art V-LLMs suffer from object hallucination, generating image descriptions inconsistent with the image content [36]. For example, a V-LLM would respond to “Where is the cat in this image?” with “The cat is on the table” when in reality there is no cat in the image. We evaluate the extent of hallucination in V-LLMs using three datasets we introduce (details in Appendix C) and the POPE dataset [36]. Our three datasets, Hal-COCO, Hal-ADE, and Hal-Act build off COCO, ADE-20K, and ActivityNet datasets respectively. The first two involve images and the latter videos. These datasets contain ‘Is there *obj* in image / video?’ type questions per sample, for two objects present and not present in the image / video. Hal-ADE object categories contain *no overlap* with COCO classes allowing evaluation on novel object categories unseen during our instruction fine-tuning. Results reported in Tab. 8 show clear improvements of LocVLM-B over baselines. We also evaluate LocVLM-B on the POPE benchmark [36] that builds

Method	VLT	Frames	Acc (%)
LLaVa (v1) [38]	✗	1	28.7
LLaVa (v1.5) [38]	✗	1	31.5
LocVLM-B	✗	1	29.2
LocVLM-L	✗	1	32.1
Video-ChatGPT [42]	✓	100	35.2
LocVLM-Vid-B	✓	100	37.4
LocVLM-Vid-B+	✓	8	38.2

Table 7. **Video VQA:** We report more results (Top-1 Accuracy) for ActivityNet-QA dataset including multiple baseline and LocVLM variants. Our proposed models exhibit top performance. VLT denotes video level training. More details in Appendix D.

Method	Hal-COCO	Hal-ADE	Hal-Act
Shikra [8]	86.2	58.7	-
LLaVa [38]	61.9	53.8	-
LocVLM-B	88.3	75.2	-
Video-ChatGPT [42]	-	-	50.6
LocVLM-Vid-B	-	-	68.7
LocVLM-Vid-B+	-	-	72.4

Table 8. **Hallucination Evaluation:** We report top-1 accuracy (%) for object presence type questions and showcase reduced object hallucination in our proposed framework.

off the COCO dataset object annotations and report results in Tab. 9. Our LocVLM showcases similar performance improvements on this dataset.

4.6. Region Description

A unique characteristic of our model (in contrast to V-LLMs like LLaVA [38] & BLIP-2 [33]) is its ability to reason with prompts involving coordinate based image space locations without any input modifications. Given a point or bounding box location, we prompt our model to generate an output describing that location. We refer to this unique ability of our model as *region description* (RD). We evaluate this RD capability of our model by generating object

Datasets	Metrics	BLIP-2	Shikra	LLaVA	Ours
Random	Accuracy (\uparrow)	88.6	86.9	50.4	87.9
	Precision (\uparrow)	84.1	94.4	50.2	83.6
	Recall (\uparrow)	95.1	79.3	99.1	93.9
	F1 Score (\uparrow)	89.3	86.2	66.6	88.5
	Yes	56.6	43.3	98.8	56.2
Popular	Accuracy (\uparrow)	82.8	84.0	49.9	86.0
	Precision (\uparrow)	76.3	87.6	49.9	79.7
	Recall (\uparrow)	95.1	79.2	99.3	93.9
	F1 Score (\uparrow)	84.7	83.2	66.4	86.3
	Yes	62.4	45.2	99.4	58.9
Adversarial	Accuracy (\uparrow)	72.1	83.1	49.7	78.8
	Precision (\uparrow)	65.1	85.6	49.9	76.6
	Recall (\uparrow)	95.1	79.6	99.1	93.7
	F1 Score (\uparrow)	77.3	82.5	66.3	84.3
	Yes	73.0	46.5	99.4	61.7

Table 9. **More object hallucination:** Results on POPE evaluation benchmark [36] indicate strong performance of our model.

Method	ZS	RefCOCO	RefCOCO+	RefCOCOg	
				Val	Test
SLR [67]	\times	-	-	-	15.4
SLR + Rerank [67]	\times	-	-	-	15.9
Kosmos-2 [45]	\times	8.67	8.82	14.3	14.1
Shikra [8]	\times	10.4	11.1	19.7	19.5
LLaVa [38]	\times	8.43	8.73	13.5	13.5
LocVLM-B	\times	14.6	15.2	26.0	26.2
Kosmos-2 [45]	\checkmark	6.34	8.25	12.4	12.2
LLava [38]	\checkmark	4.23	7.26	10.6	10.3
LocVLM-B	\checkmark	11.0	11.1	20.6	20.7

Table 10. **Region Description:** We report METEOR scores for RD task [45]. Test-B split is used for RefCOCO & RefCOCO+ datasets. Our method outperforms all prior work.

level descriptions focused on contextual information (e.g. surrounding of that object in the image). Following evaluation protocol in [45] for region description, we extend their evaluation to three standard referring localization datasets from RefCOCO [29] and report these results in Tab. 10. We select the METEOR score as the evaluation metric to account for variations in word choice in generated answers which may be acceptable in various cases (e.g. different sentence structure leading to alternate word ordering). Our results indicate clear improvements over the LLaVA baseline [38] as well as prior state-of-the-art. We attribute these improvements to our pseudo-data based training.

4.7. Ablations

Next we conduct ablative studies on separate components of our proposed setup: IFT objectives, location type, and pseudo-data. We follow the same training strategy as described in Sec. 4.1 and present these results in Tab. 11. LocVLM-B is used for all these experiments. The significance of each IFT objective is verified in Tab. 11 (top) with

LocPred	NegPred	RevLoc	GQA	RD	A-QA
\times	\times	\times	44.7	10.3	35.2
\checkmark	\times	\times	45.2	12.2	35.8
\checkmark	\checkmark	\times	46.9	12.5	37.2
\checkmark	\checkmark	\checkmark	47.3	20.7	37.4
Location Type		PD	GQA	RD	A-QA
Point		\checkmark	47.3	20.6	37.4
Bounding Box		\checkmark	47.3	20.7	37.4
Bounding Box		\times	46.5	11.6	37.1

Table 11. **Ablations:** We report top-1 accuracy (%) on GQA and ActivityNet-QA (A-QA) datasets and METEOR scores for RD task on RefCOCOg test split. (top) We ablate proposed instruction fine-tuning objectives to verify usefulness of each objective. (bottom) We first ablate point based and bounding box based location forms to showcase minimal difference across them. We next ablate use of object description pseudo-data (PD). We highlight the improvements due to pseudo-data, especially on the RD task.

LocPred	NegPred	RevLoc	A-QA
\times	\times	\times	37.4
\checkmark	\times	\times	37.6
\checkmark	\checkmark	\times	38.2
\checkmark	\checkmark	\checkmark	38.2

Table 12. **Video Ablation:** We report top-1 accuracy (%) on ActivityNet-QA (A-QA) dataset. Results indicate the generality of proposed IFT objectives for video domain training as well.

consistent performance improvements across tasks. The generality of our approach to differing location type (i.e. points vs bounding boxes) and usefulness of pseudo-data is visible in Tab. 11 (bottom). In particular, we highlight the notable performance improvement for RD task gained from using pseudo-data. We also conduct ablations for our video domain training setup and report these results in Tab. 12. The LocVLM-Vid-B+ variant is used in these experiments. Our results showcase the usefulness of proposed IFT objectives for video domain learning as well.

5. Conclusion

We introduce a simple framework that equips visual-LLMs (V-LLMs) with greater spatial understanding, termed LocVLM. We leverage the idea of encoding image coordinates within language to propose three instruction fine-tuning (IFT) objectives. This training process endows V-LLMs with the ability to reason about spatial composition of images using image space coordinates within text. A data efficient training pipeline utilizing pseudo-data allows our approach to achieve state-of-the-art results in Image VQA, Video VQA, and Region Description while improving spatial awareness and reducing object hallucination.

References

- [1] Jake K. Aggarwal and Michael S. Ryoo. Human activity analysis. *ACM Computing Surveys (CSUR)*, 43:1 – 43, 2011. [7](#)
- [2] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *arXiv preprint arXiv:2204.14198*, 2022. [2](#), [7](#)
- [3] Anas Awadalla, Irena Gao, Joshua Gardner, Jack Hessel, Yusuf Hanafy, Wanrong Zhu, Kalyani Marathe, Yonatan Bitton, Samir Gadre, Jenia Jitsev, Simon Kornblith, Pang Wei Koh, Gabriel Ilharco, Mitchell Wortsman, and Ludwig Schmidt. Openflamingo, 2023. [2](#)
- [4] Pratyay Banerjee et al. Weakly supervised relative spatial reasoning for visual question answering. *ICCV*, 2021. [6](#)
- [5] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. [1](#), [2](#)
- [6] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, pages 213–229. Springer, 2020. [2](#), [3](#)
- [7] Soravit Changpinyo, Piyush Kumar Sharma, Nan Ding, and Radu Soricut. Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3557–3567, 2021. [5](#)
- [8] Keqin Chen, Zhao Zhang, Weili Zeng, Richong Zhang, Feng Zhu, and Rui Zhao. Shikra: Unleashing multi-modal llm’s referential dialogue magic. *arXiv preprint arXiv:2306.15195*, 2023. [1](#), [2](#), [3](#), [6](#), [7](#), [8](#)
- [9] Shoufa Chen, Peize Sun, Yibing Song, and Ping Luo. Diffusiondet: Diffusion model for object detection. *arXiv preprint arXiv:2211.09788*, 2022. [2](#), [3](#)
- [10] Ting Chen, Saurabh Saxena, Lala Li, David J Fleet, and Geoffrey Hinton. Pix2seq: A language modeling framework for object detection. *arXiv preprint arXiv:2109.10852*, 2021. [2](#), [3](#), [1](#)
- [11] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, 2023. [1](#), [2](#)
- [12] Jaemin Cho et al. Dall-eval: Probing the reasoning skills and social biases of text-to-image generation models. *ICCV*, 2023. [2](#)
- [13] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022. [2](#)
- [14] Yufeng Cui, Lichen Zhao, Feng Liang, Yangguang Li, and Jing Shao. Democratizing contrastive language-image pre-training: A clip benchmark of data, model, and supervision. *arXiv preprint arXiv:2203.05796*, 2022. [2](#)
- [15] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. Instructclip: Towards general-purpose vision-language models with instruction tuning. *arXiv preprint arXiv:2305.06500*, 2023. [7](#)
- [16] Jian Ding, Nan Xue, Guisong Xia, and Dengxin Dai. Decoupling zero-shot semantic segmentation. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11573–11582, 2021. [2](#)
- [17] Zi-Yi Dou, Aishwarya Kamath, Zhe Gan, Pengchuan Zhang, Jianfeng Wang, Linjie Li, Zicheng Liu, Ce Liu, Yann LeCun, Nanyun Peng, Jianfeng Gao, and Lijuan Wang. Coarse-to-fine vision-language pre-training with fusion in the backbone. *ArXiv*, abs/2206.07643, 2022.
- [18] Golnaz Ghiasi, Xiuye Gu, Yin Cui, and Tsung-Yi Lin. Open-vocabulary image segmentation. In *ECCV*, 2022. [2](#)
- [19] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015. [2](#), [3](#)
- [20] Gokhale et al. Benchmarking spatial relationships in text-to-image generation, 2022. [2](#)
- [21] Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. Zero-shot detection via vision and language knowledge distillation. *arXiv e-prints*, pages arXiv–2104, 2021. [2](#)
- [22] Tanmay Gupta and Aniruddha Kembhavi. Visual programming: Compositional visual reasoning without training. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14953–14962, 2022. [2](#)
- [23] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 961–970, 2015. [5](#)
- [24] Joy Hsu et al. What’s left? concept grounding with logic-enhanced foundation models. *NeurIPS*, 2023. [2](#)
- [25] Drew A. Hudson and Christopher D. Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6693–6702, 2019. [3](#)
- [26] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V. Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning*, 2021. [1](#)
- [27] Aishwarya Kamath, Mannat Singh, Yann LeCun, Gabriel Synnaeve, Ishan Misra, and Nicolas Carion. Mdetrmultimodal detection for end-to-end multi-modal understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1780–1790, 2021. [2](#)
- [28] Amita Kamath et al. What’s ”up” with vision-language models? investigating their struggle with spatial reasoning. *EMNLP*, 2023. [2](#)

- [29] Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. Referitgame: Referring to objects in photographs of natural scenes. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 787–798, 2014. 8
- [30] Xin Lai, Zhuotao Tian, Yukang Chen, Yanwei Li, Yuhui Yuan, Shu Liu, and Jiaya Jia. Lisa: Reasoning segmentation via large language model. *arXiv preprint arXiv:2308.00692*, 2023. 2
- [31] Boyi Li, Kilian Q Weinberger, Serge Belongie, Vladlen Koltun, and René Ranfl. Language-driven semantic segmentation. *ICLR*, 2022. 2
- [32] Jiahao Li, Greg Shakhnarovich, and Raymond A. Yeh. Adapting clip for phrase localization without further training. *ArXiv*, abs/2204.03647, 2022. 2
- [33] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023. 1, 2, 3, 6, 7
- [34] Kunchang Li, Yanan He, Yi Wang, Yizhuo Li, Wenhai Wang, Ping Luo, Yali Wang, Limin Wang, and Yu Qiao. Videochat: Chat-centric video understanding. *arXiv preprint arXiv:2305.06355*, 2023. 7
- [35] Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, Kai-Wei Chang, and Jianfeng Gao. Grounded language-image pre-training. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10955–10965, 2021. 2
- [36] Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. *arXiv preprint arXiv:2305.10355*, 2023. 7, 8, 2
- [37] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 5, 2, 3
- [38] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *arXiv preprint arXiv:2304.08485*, 2023. 1, 2, 3, 4, 5, 6, 7, 8
- [39] Huaishao Luo, Junwei Bao, Youzheng Wu, Xiaodong He, and Tianrui Li. Segclip: Patch aggregation with learnable centers for open-vocabulary semantic segmentation. *ArXiv*, abs/2211.14813, 2022. 1, 2
- [40] Jitendra Malik. Visual grouping and object recognition. In *Proceedings 11th International Conference on Image Analysis and Processing*, pages 612–621. IEEE, 2001. 2
- [41] David Marr. *Vision: A computational investigation into the human representation and processing of visual information*. MIT press, 1982. 1
- [42] Salman Khan Muhammad Maaz, Hanoona Rasheed and Fahad Khan. Video-chatgpt: Towards detailed video understanding via large vision and language models. *ArXiv 2306.05424*, 2023. 2, 5, 6, 7, 3
- [43] Jishnu Mukhoti, Tsung-Yu Lin, Omid Poursaeed, Rui Wang, Ashish Shah, Philip H. S. Torr, and Ser Nam Lim. Open vocabulary semantic segmentation with patch aligned contrastive learning. *CVPR*, abs/2212.04994, 2023. 1, 2
- [44] OpenAI. GPT-4 technical report. <https://arxiv.org/abs/2303.08774>, 2023. 1, 2
- [45] Zhiliang Peng, Wenhui Wang, Li Dong, Yaru Hao, Shaohan Huang, Shuming Ma, and Furu Wei. Kosmos-2: Grounding multimodal large language models to the world. *arXiv preprint arXiv:2306.14824*, 2023. 2, 8
- [46] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 1, 2, 3
- [47] Kanchana Ranasinghe and Michael S. Ryoo. Language-based action concept spaces improve video self-supervised learning. In *NeurIPS*, 2023. 2
- [48] Kanchana Ranasinghe, Brandon McKinzie, Sachin Ravi, Yinfei Yang, Alexander Toshev, and Jonathon Shlens. Perceptual grouping in contrastive vision-language models. In *ICCV*, 2023. 1, 2
- [49] Kanchana Ranasinghe, Xiang Li, Kumara Kahatapitiya, and Michael S. Ryoo. Understanding long videos in one multimodal language model pass, 2024. 2
- [50] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. 2, 3
- [51] Michael S. Ryoo and Jake K. Aggarwal. Recognition of composite human activities through context-free grammar based representation. *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, 2:1709–1718, 2006. 7
- [52] Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*, 2022. 2
- [53] D’idac Sur’is, Sachit Menon, and Carl Vondrick. Vipergpt: Visual inference via python execution for reasoning. *ArXiv*, abs/2303.08128, 2023. 2
- [54] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9627–9636, 2019. 2, 3
- [55] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 1, 2, 3
- [56] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. 1, 2, 3
- [57] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *IJCV*, 104(2):154–171, 2013. 2

- [58] Wenhai Wang, Zhe Chen, Xiaokang Chen, Jiannan Wu, Xizhou Zhu, Gang Zeng, Ping Luo, Tong Lu, Jie Zhou, Yu Qiao, et al. Visionllm: Large language model is also an open-ended decoder for vision-centric tasks. *arXiv preprint arXiv:2305.11175*, 2023. [2](#), [3](#), [1](#)
- [59] Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. Finetuned language models are zero-shot learners. *ArXiv*, abs/2109.01652, 2021. [2](#)
- [60] Jiarui Xu, Shalini De Mello, Sifei Liu, Wonmin Byeon, Thomas Breuel, Jan Kautz, and Xiaolong Wang. Groupvit: Semantic segmentation emerges from text supervision. *CVPR*, 2022. [2](#)
- [61] Jilan Xu, Junlin Hou, Yuejie Zhang, Rui Feng, Yi Wang, Yu Qiao, and Weidi Xie. Learning open-vocabulary semantic segmentation models from natural language supervision. *ArXiv*, abs/2301.09121, 2023. [2](#)
- [62] Shen Yan, Tao Zhu, Zirui Wang, Yuan Cao, Mi Zhang, Soham Ghosh, Yonghui Wu, and Jiahui Yu. Videococa: Video-text modeling with zero-shot transfer from contrastive captioners. *arXiv*, 2022. [7](#)
- [63] Antoine Yang, Antoine Miech, Josef Sivic, Ivan Laptev, and Cordelia Schmid. Just ask: Learning to answer questions from millions of narrated videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1686–1697, 2021. [7](#)
- [64] Antoine Yang, Antoine Miech, Josef Sivic, Ivan Laptev, and Cordelia Schmid. Zero-shot video question answering via frozen bidirectional language models. *arXiv preprint arXiv:2206.08155*, 2022. [7](#)
- [65] Lewei Yao, Runhui Huang, Lu Hou, Guansong Lu, Minzhe Niu, Hang Xu, Xiaodan Liang, Zhenguo Li, Xin Jiang, and Chunjing Xu. FILIP: Fine-grained interactive language-image pre-training. In *ICLR*, 2022. [2](#)
- [66] Haoxuan You, Haotian Zhang, Zhe Gan, Xianzhi Du, Bowen Zhang, Zirui Wang, Liangliang Cao, Shih-Fu Chang, and Yinfei Yang. Ferret: Refer and ground anything anywhere at any granularity. *ArXiv*, abs/2310.07704, 2023. [2](#)
- [67] Licheng Yu, Hao Tan, Mohit Bansal, and Tamara L. Berg. A joint speaker-listener-reinforcer model for referring expressions. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3521–3529, 2016. [8](#)
- [68] Yuhang Zang, Wei Li, Jun Han, Kaiyang Zhou, and Chen Change Loy. Contextual object detection with multimodal large language models. *arXiv preprint arXiv:2305.18279*, 2023. [2](#)
- [69] Yan Zeng, Xinsong Zhang, and Hang Li. Multi-grained vision language pre-training: Aligning texts with visual concepts. *ArXiv*, abs/2111.08276, 2021. [2](#)
- [70] Haotian Zhang, Pengchuan Zhang, Xiaowei Hu, Yen-Chun Chen, Liunian Harold Li, Xiyang Dai, Lijuan Wang, Lu Yuan, Jenq-Neng Hwang, and Jianfeng Gao. Glipv2: Unifying localization and vision-language understanding. *ArXiv*, abs/2206.05836, 2022. [2](#)
- [71] Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding. *ArXiv*, abs/2306.02858, 2023. [7](#)
- [72] Renrui Zhang, Jiaming Han, Aojun Zhou, Xiangfei Hu, Shilin Yan, Pan Lu, Hongsheng Li, Peng Gao, and Yu Jiao Qiao. Llama-adapter: Efficient fine-tuning of language models with zero-init attention. *ArXiv*, abs/2303.16199, 2023. [7](#)
- [73] Shilong Zhang, Peize Sun, Shoufa Chen, Min Xiao, Wenqi Shao, Wenwei Zhang, Kai Chen, and Ping Luo. Gpt4roi: Instruction tuning large language model on region-of-interest. *arXiv preprint arXiv:2307.03601*, 2023. [2](#)
- [74] Yabo Zhang, Zihao Wang, Jun Hao Liew, Jingjia Huang, Manyu Zhu, Jiashi Feng, and Wangmeng Zuo. Associating spatially-consistent grouping with text-supervised semantic segmentation. *ArXiv*, abs/2304.01114, 2023. [1](#), [2](#)
- [75] Yang Zhao, Zhijie Lin, Daquan Zhou, Zilong Huang, Jiashi Feng, and Bingyi Kang. Bubogpt: Enabling visual grounding in multi-modal llms. *arXiv preprint arXiv:2307.08581*, 2023. [2](#)
- [76] Chong Zhou, Chen Change Loy, and Bo Dai. Extract free dense labels from clip. In *European Conference on Computer Vision*, 2021. [2](#)
- [77] Xueyan Zou, Jianwei Yang, Hao Zhang, Feng Li, Linjie Li, Jianfeng Gao, and Yong Jae Lee. Segment everything everywhere all at once. In *NeurIPS*, 2023. [5](#), [2](#)

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 $n_a = 24^2$ 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 $p, q = 0, 1, \dots, 13$. 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 processed 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 prepend 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 if the terms `left` or `right` are present in the generated outputs, and rate it a success if 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 reasonings.

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 us 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) are 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).

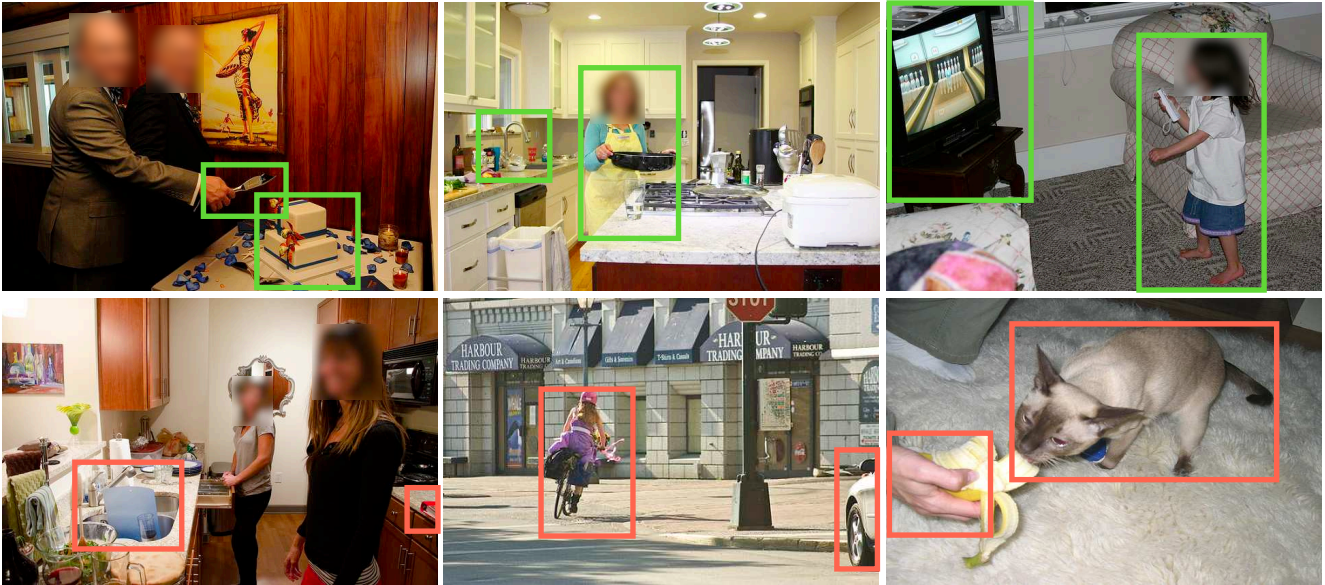


Figure 3. Visualizing Spatial Reasoning: We illustrate example images on which we perform our toy experiment for spatial reasoning (Appendix E). Success cases on top row (green) and failure cases on bottom row (red).



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.

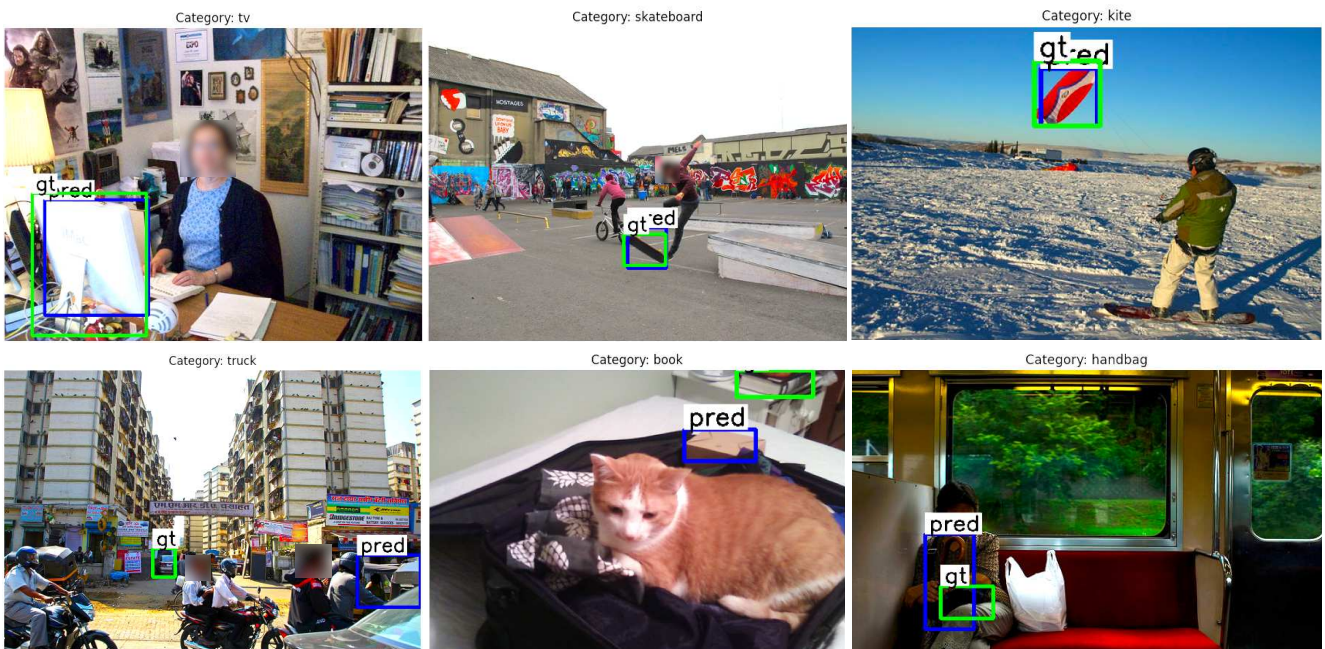


Figure 5. Visualization of LocPred Objective: We illustrate the bounding box locations generated by our framework (blue) when queried with a category label (top of each image) and compare with the ground-truth bounding boxes (green). Success cases on top and failure cases on bottom.