Jack of All Tasks, Master of Many: Designing General-purpose Coarse-to-Fine Vision-Language Model (Supplementary Material)

Axis	Metric and Split
COCO Cap	CIDEr on Karpathy test
VQAv2	Accuracy on val
VCR	Accuracy on val in Q → AR setup
POPE	F1 score on Random split
HM	Accuracy on test
TextVQA	Accuracy on test
REC	Precision@IoU=0.5 on RefCOCO val
RES	mIoU on RefCOCO val
GREC	Precision on RefCOCO val
GRES	gIoU on RefCOCO val
BoxQA	Accuracy on Visual7W
NLVR2	Accuracy on dev
IconQA	Accuracy on test
iCoSeg	Average Jaccard index ($\mathcal{J}$) on test

Table A.1. **Details of the reported metrics and split information in every axis of the radar plot in Figure 1.** Red: Single-image coarse-level tasks, Blue: Single-image region-level tasks, Olive-Green: Multi-image coarse-level tasks, and Plum: Multi-image region-level tasks.

A. Radar Chart Figure 1 Details

In this section, we explain the details of the radar chart in Figure 1, which summarizes the comparative performance of VistaLLM with MiniGPT-v2 [5], Ferret [37], Shikra [6] and GPT4RoI [40]. None of these baselines address segmentation and multi-image tasks using a single framework. First, for illustrative purposes, we normalize each axis by the score achieved by VistaLLM, which turns the axes in the range $(0, 1]$. Next, we choose the origin of each axes suitably to distinctly separate the the inner and outer frames for better readability. For BoxQA, REC, and COCO Cap, the origin is at 0.97, 0.96, and 0.75 normalized values, respectively. For all remaining axes, the origin is at 0.92 normalized value. Finally, we annotate each vertex with absolute performance metric scores. The reported metric and split name for each axis are listed in Table A.1.

B. Adaptive Sampling Algorithm

The algorithm of the proposed gradient-aware adaptive sampling technique is given in Algorithm 1. Section 3.2 of the main manuscript provides details of each step.

Algorithm 1 Gradient-aware Adaptive Sampling

```
Require: Mask contour $\mathcal{C}$
  Number of dense points $M$
  Final number of sampling points $N$ (N ≪ M)
  $[p_1, \ldots, p_{\mathrm{M}}] \leftarrow$ Uniform-Sample($\mathcal{C}$)     ▷ Contour Discretization
  for $i \in \{1, \ldots, M\}$ do
      $\vec{l_1}$ = Join($p_{\mathrm{i}}, p_{\mathrm{i}-1}$)
      $\vec{l_2}$ = Join($p_{\mathrm{i}-1}, p_{\mathrm{i}+1}$)
      $\theta_{\mathrm{i}} = \angle \vec{l_1} \vec{l_2}$                 ▷ Gradient Calculation
  end for
  $\text{Final}_{\text{points}} \leftarrow$ []
  indices $\leftarrow$ argsort($\theta_{\mathrm{i} \in \{1, \ldots, M\}}$)[M-N:]     ▷ Sorting
  for $j \in$ indices do
      $p_{\mathrm{j}} \leftarrow$ Quantize($p_{\mathrm{j}}$)
      AddItem($\text{Final}_{\text{points}}, p_{\mathrm{j}}$)           ▷ Quantization
  end for
  $\text{Final}_{\text{points}}$ is the final list of sampled points.
```

C. VistaLLM vs Existing Region-level MLLMs

With the fast progress of region-level general-purpose vision systems, works such as GPT4RoI [40], Shikra [6], VisionLLM [32], KOSMOS-2 [26] and Ferret [37] resemble VistaLLM, as they also aim to unify tasks with different granularity in a unified system. Additional related works in this category includes PVIT [4], COMM [12], CogVLM [33] and MiniGPT-v2 [5]. Moreover, methods like Visual ChatGPT [35], BuboGPT [41], DetGPT [27], and LISA [15] employ external additional detection and segmentation modules to unify fine-grained tasks in a two-stage approach. Nevertheless, there exist clear differences between VistaLLM from existing methods. First, we present the first general-purpose system to support all possible input and output formats, e.g., multiple images, natural language, coordinate points, bounding boxes, segmentation masks as inputs, and free-flowing text, points, boxes, and masks as output. Table C.1 shows a side-by-side comparison of input-output formats of all existing baselines. While Ferret supports boxes, points, and masks in the input, it can not generate a mask as output and, hence, can not address the segmentation task. On the

	Model	Input Type					Output Type			
		Multiple Images	Text	Points	Boxes	Masks	Text	Points	Boxes	Masks
Two-Stage	Visual ChatGPT [35]	✗	✓	✗	✗	✗	✓	✗	✓	✓
	BuboGPT [41]	✗	✓	✗	✗	✗	✓	✗	✓	✗
	DetGPT [27]	✗	✓	✗	✗	✗	✓	✗	✓	✗
	LISA [15]	✗	✓	✗	✗	✗	✓	✗	✗	✓
End-to-End	LLaVa [20]	✗	✓	✗	✗	✗	✓	✗	✗	✗
	InstructBLIP [9]	✗	✓	✗	✗	✗	✓	✗	✗	✗
	GPT4RoI [40]	✗	✓	✗	✓	✗	✓	✗	✗	✗
	KOSMOS-2 [26]	✗	✓	✗	✓	✗	✓	✗	✓	✗
	VisionLLM [32]	✗	✓	✗	✗	✗	✓	✗	✓	✓
	Shikra [6]	✗	✓	✓	✓	✗	✓	✓	✓	✗
	PVIT [4]	✗	✓	✗	✓	✗	✓	✗	✗	✗
	CogVLM [33]	✗	✓	✗	✓	✗	✓	✗	✓	✗
	COMM [12]	✗	✓	✗	✓	✗	✓	✗	✓	✗
	MiniGPT-v2 [5]	✗	✓	✓	✓	✗	✓	✓	✓	✗
	Ferret [37]	✗	✓	✓	✓	✓	✓	✓	✓	✗
	VistaLLM	✓	✓	✓	✓	✓	✓	✓	✓	✓

Table C.1. **Comparison of VistaLLM vs. existing general-purpose vision systems regarding input and output types.** VistaLLM supports all possible formats, including multiple images, natural language, points, bounding boxes, segmentation masks as inputs, and free-flowing text, points, boxes, and masks as output.

	Model	Image-level Tasks			Region-level Tasks					
		Single-image		Multi-image	Single-image					Multi-image
		VQAv2 & Captioning	Reasoning	Reasoning	BoxQA	PointQA	Detection	Segmentation	Multi-instance Segmentation	CoSeg
Two-Stage	Visual ChatGPT [35]	✓	✓	✗	✗	✗	✓	✓	✓	✗
	BuboGPT [41]	✓	✓	✗	✗	✗	✓	✗	✗	✗
	DetGPT [27]	✓	✓	✗	✗	✗	✓	✗	✗	✗
	LISA [15]	✓	✓	✗	✗	✗	✗	✓	✗	✗
End-to-End	LLaVa [20]	✓	✓	✗	✗	✗	✗	✗	✗	✗
	InstructBLIP [9]	✓	✓	✗	✗	✗	✗	✗	✗	✗
	GPT4RoI [40]	✓	✓	✗	✓	✗	✗	✗	✗	✗
	KOSMOS-2 [26]	✓	✓	✗	✓	✗	✓	✗	✗	✗
	VisionLLM [32]	✓	✓	✗	✗	✗	✓	✓	✓	✗
	Shikra [6]	✓	✓	✗	✓	✓	✓	✗	✗	✗
	PVIT [4]	✓	✓	✗	✓	✗	✗	✗	✗	✗
	CogVLM [33]	✓	✓	✗	✓	✗	✓	✗	✗	✗
	COMM [12]	✓	✓	✗	✓	✗	✓	✗	✗	✗
	MiniGPT-v2 [5]	✓	✓	✗	✓	✓	✓	✗	✗	✗
	Ferret [37]	✓	✓	✗	✓	✓	✓	✗	✗	✗
	VistaLLM	✓	✓	✓	✓	✓	✓	✓	✓	✓

Table C.2. **Comparison of VistaLLM vs. existing general-purpose vision systems regarding supported tasks.** VistaLLM integrates a wide range of image-level and region-level vision-language reasoning and grounding tasks over single and multiple input images into a unified framework.

other hand, VisionLLM can solve segmentation but cannot process points, boxes, and masks in input and can not solve REG, BoxQA, and PointQA. Second, unlike all existing works, VistaLLM supports multi-image input, enabling us to reason and ground over more than one image and solve tasks like NLVR and CoSeg. Our proposed instruction-guided image tokenizer module refines and compresses the global image embeddings of multiple images, helping VistaLLM to filter the necessary visual information required for the current task. Table C.2 systematically illustrates the capability of VistaLLM to solve a wide range of image-level and region-level tasks over single and multiple input images compared to previous systems. Third, to efficiently convert segmentation masks into sequences, we propose a gradient-aware adaptive contour sampling scheme, which improves over previously used uniform sampling approach [7, 8, 22, 42] by $3-4$ mIoU scores on different segmentation benchmarks. Lastly, we collect a new training benchmark CoinIt, containing 6.8M training samples and propose a new task, AttCoSeg (**Att**ribute-level **Co-Seg**mentation) which addresses the lack of publicly-available multi-image region-level datasets. Our proposed system achieves stronger performance across 15 different evaluation benchmarks, including mitigating object hallucination to a significant extent.

D. Dataset Details

This section provides additional details of our training and evaluation datasets.

COCO Captioning: Captions for the COCO dataset [18] were sourced from Amazon's Mechanical Turk (AMT), with workers adhering to specified guidelines to ensure caption quality. The dataset includes 330,000 images, divided into training, validation, and test categories. These categories comprise 413,915 captions for 82,783 images in training, 202,520 captions for 40,504 images in validation, and 379,249 captions for 40,775 images in the test set.

VQAv2: VQAv2 dataset [1] contains a collection of over 200,000 images, each paired with a portion of the more than 1.1 million questions asked, gathering in total over 11 million responses. The questions cover a wide range, from simple yes/no and counting queries to more complex open-ended ones.

RefCOCO & RefCOCO+: The RefCOCO and RefCOCO+ datasets [21] were created through a two-player game mechanism [38]. RefCOCO features 142,209 descriptive expressions for 50,000 objects across 19,994 images, whereas RefCOCO+ includes 141,564 expressions for 49,856 objects in 19,992 images. Both datasets are divided into training, validation, and two test sets – Test A and Test B. Test A focuses on images with multiple people. At the same time, Test B features images with multiple instances of all other objects. A key difference between the two datasets is that RefCOCO+ excludes location words from its expressions, making it more complex than RefCOCO. We perform referring expression comprehension (REC) and referring expression segmentation (RES) tasks on the RefCOCO and RefCOCO+ datasets.

RefCOCOg: The RefCOCOg dataset was assembled using Amazon Mechanical Turk, where participants were tasked with crafting natural language descriptions for objects. It comprises 85,474 expressions for 54,822 objects in 26,711 images. Notably, the expressions in RefCOCOg are longer and more intricate, averaging 8.4 words, in contrast to the more concise expressions in RefCOCO and RefCOCO+, which average 3.5 words. This complexity makes RefCOCOg a more challenging dataset. We utilize the UMD partition [25] of RefCOCOg, as it provides both validation and testing sets, and there is no overlap between training and validation images. We address both REC and RES tasks on RefCOCOg.

gRefCOCO: The gRefCOCO dataset [11, 19] empowers generalized referring expression comprehension (GREC) and generalized referring expression segmentation (GRES) tasks, which address the limitations of classical REC and RES problem where there is always one target object. In contrast, GREC and GRES allow expressions to refer to an arbitrary number of target objects, including multi-target and no-target scenarios, and help bring referring segmentation into more realistic scenarios with advanced usages. The gRefCOCO dataset contains 278,232 expressions, including 80,022 multi-target and 32,202 no-target expressions, referring to 60,287 distinct instances in 19,994 images. Masks and bounding boxes for all target instances are given. Some of the single-target expressions of gRofCOCO are inherited from RefCOCO. We perform both GREC and GRES using the gRefCOCO dataset.

Flickr: The Flickr30K Entities dataset [28] is a pioneering collection in the field of grounded captioning. It includes 31,783 images paired with 158,000 caption annotations. Each caption is carefully annotated, linking every noun phrase to a manually outlined referential bounding box. The dataset features a total of 276,000 such annotated bounding boxes, offering a rich resource for image and language processing research. We use Flickr dataset during training for spot captioning task, where we instruct the model to generate a caption of the input image, and locate all the objects in the images by drawing bounding boxes.

Visual Genome: The Visual Genome dataset [14] is a key resource for understanding the complex relationships within images. It contains over 100,000 images, with each image extensively annotated to capture an average of 21 objects, 18 attributes, and 18 inter-object relationships. A distinctive feature of this dataset is the alignment of objects, attributes, relationships, and region descriptions with the standardized WordNet terminologies. This alignment makes it particularly useful for tasks like Region Description and Entity Recognition. Each annotated region in the dataset is accompanied by descriptive text, providing a wealth of data for image understanding and semantic modeling. For referring expression generation (REG) purposes, we utilize a subset of this dataset, which includes around 180,138 region-caption pairs.

VCR: The Visual Commonsense Reasoning (VCR) dataset [39] contains 290,000 multiple-choice questions derived from 110,000 movie scenes. Each scene is paired with a question demanding common-sense reasoning, an answer, and a rationale for that answer. The unique aspect of VCR is its requirement for models to not only provide answers to complex visual questions but also to explain their reasoning. This dataset encompasses two sub-tasks: Question Answering (Q → A), where the model selects the correct answer from four options, and answer justification (QA → R), where the model, given a question and its correct answer, must choose the most fitting rationale from four options. Model performance in VCR is assessed using the Q → AR metric, which measures the accuracy of both answering questions and providing the correct justifications.

LLaVa: The LLaVA-Instruct-150K[1] [20] is a collection of 158K unique language-image instruction-following samples in total, including 58K in conversations, 23K in the detailed description, and 77k in complex reasoning, respectively. We incorporate the LLaVa dataset during the training of our model.

LookTwiceQA: The LookTwiceQA [24] dataset contains two different tasks - PointQA and BoxQA. The questions are in three different templates - (i) What color is this [region]? (ii) What shape is this [region]? and (iii) What action is this [region] doing? The question contains either an input point or a box with three different granularity of objects - any object, superclass, and object class. The train set contains 40,409 questions across 12,867 images, and the test-dev set contains 5,673 questions across 1,838 images.

Visual7W: The Visual7W dataset [43] is primarily tailored for Visual Question Answering (VQA) tasks, featuring a specialized dataset for region-level QA. In Visual7W, models encounter an image paired with a "which"-type question, for instance, "Which one is the orange in the fruit basket?". Participants are provided with four bounding boxes in the image and must choose the correct one as the answer. The Visual7W dataset comprises 25,733 images and 188,068 such questions.

TextVQA: TextVQA [30] is a QA dataset containing 45,336 questions based on 28,408 images, designed to challenge models in detecting, interpreting, and reasoning about text present in images to generate accurate answers. We use the TestVQA dataset as an unseen evaluation benchmark.

IconQA: IconQA [23] measures models' abstract diagram understanding and comprehensive cognitive reasoning abilities. We use the test set of its multi-text-choice task, containing 6,316 samples, as an unseen evaluation benchmark.

Hateful Memes (HM): The hateful memes dataset [13], containing more than 10,000 image samples, is a binary classification dataset to justify whether a meme contains hateful content. The memes were selected in such a way that strictly unimodal classifiers would struggle to classify them correctly. We use the HM dataset as an unseen evaluation benchmark.

POPE: The POPE evaluation benchmark [17] evaluates the sevearity of object hallucination problem in MLLMs. POPE consists of three different test splits - popular, random, and adversarial- containing around 3,000 samples. Given an image and a question, "Is there a <object> in the image?" the model has to answer with 'yes' or 'no.'

NLVR2: The Natural Language for Visual Reasoning (NLVR2) corpora, containing 107,292 samples, determine whether a sentence is true about a pair of input images. The data was collected through crowdsourcing, and solving the task requires reasoning about sets of objects, comparisons, and spatial relations.

CoSeg: We use three datasets for object co-segmentation task - PASCAL VOC2010 [10], MSRC [34] and iCoSeg [2]. PASCAL contains a total of 1,037 images of 20 object classes. MSRC includes seven classes: bird, car, cat, cow, dog, plane, and sheep. Each class contains ten images. iCoseg dataset consists of 643 images from 38 categories. Large variances of viewpoints and deformations are present in this dataset.

AttCoSeg: Since the existing object co-segmentation datasets [2, 10, 34] are small-scale and simple to solve, we construct a more challenging larger-scale multi-image region-level dataset. We use Group-wise RES [36] annotations to sample high-quality images containing objects with similar fine-grained attributes (shape, color, size, position). We refer to such images as positives. While training VistaLLM, we input these positive image pairs and ask the model to segment the object with common traits in both of them. We name this task attribute-level co-segmentation (AttCoSeg), which contains over 804k training samples, and help VistaLLM to gain significant generalized reasoning and grounding ability over multiple input images.

E. Examples Instructions for Different Tasks

Section 5.1 discusses transforming public datasets like REC, RES, GREC, and GRES into instruction-following format by employing meticulously crafted task templates. These templates are detailed in Table E.1. We have included only 2-3 examples for each task for brevity. We manually write one example description of each task and resort to GPT-3.5 [3] to create hundreds of variations. During training, we randomly pick one instruction for each sample.

F. Additional Ablation Study

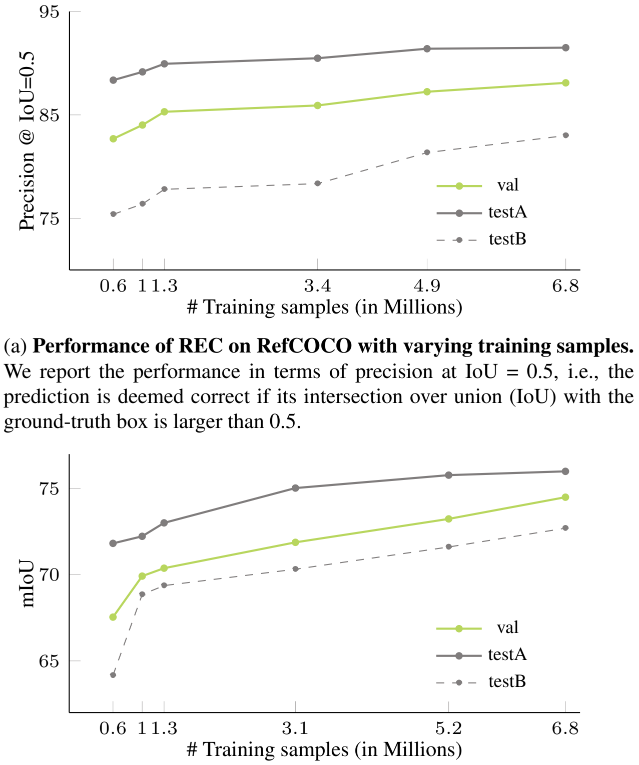
In this section, we conduct additional ablation experiments on training dataset, and the image encoder.

Size of training dataset: We study the effect of increasing training samples for REC and RES tasks in Figure F.1. We start with REC and REG training datasets for the REC task in Figure F.1a, resulting in 0.6M training samples. We

[1] https://huggingface.co/datasets/liuhaotian/LLaVA-Instruct-150K

Task	Example Instructions
Captioning	• Can you give me a brief description of this image <image>? • Give me a short description of the picture <image>. • What's happening in the image <image> at a glance?
VQAv2	• Looking at the image <image>, can you quickly answer my question: <question>. • After examining the image <image>, can you provide a brief response to the following question: <question>. • Considering the image <image>, please provide a straightforward answer to <question>.
REC	• Locate the object described by <expr> in <image>. There's just one specific object. Provide the outcome using the $[x_0, y_0, x_1, y_1]$ arrangement, showing the upper-left and lower-right box positions. • Find the location of the item referenced in <expr> within <image>. We're referring to a single item. Output the result in $[x_0, y_0, x_1, y_1]$ arrangement, showing the upper-left and lower-right bounding box corners.
RES	• Tell me where <expr> is located in <image>. There's only one object. Provide the coordinates of 32 points on the object's outline. Present the result in $[x_0, y_0, x_1, y_1, ..., x_{31}, y_{31}]$ format. • What is <expr>'s location within <image>? There's just one thing to consider. Share the coordinates of 32 uniform points on the object's edge. Show it in $[x_0, y_0, x_1, y_1, ..., x_{31}, y_{31}]$ format.
GREC	• Recognize all objects indicated by <expr> in <image>. If no object is located, return an empty string. If one or more objects are located, output the bounding boxes as $[x_0, y_0, x_1, y_1]$, indicating the top-left and bottom-right corner points. Use <bsep> to differentiate multiple bounding boxes. • Pinpoint all items referenced by <expr> in <image>. If no object is detected, return an empty string. If one or more target objects are found, provide the bounding boxes as $[x_0, y_0, x_1, y_1]$, signifying the top-left and bottom-right corner points. Use <bsep> to separate multiple bounding boxes.
GRES	• Find all items indicated by <expr> within <image>. If no target object is recognized, produce an empty string. If one or more target objects are identified, output the coordinates of 32 points along each object's contour. Display each object mask in $[x_0, y_0, x_1, y_1, ..., x_{31}, y_{31}]$ format. Use <msep> to distinguish multiple objects. • Recognize all referenced items via <expr> in <image>. If no target object is found, generate an empty string. If one or more target objects are found, present the coordinates of 32 points along each object's edge. Show each object mask in $[x_0, y_0, x_1, y_1, ..., x_{31}, y_{31}]$ format. Utilize <msep> to distinguish multiple objects.
REG	• Please generate a unique description for the area <objs> displayed in the image <image>. • What can you tell me about the area <objs> in the image <image> that sets it apart from the rest? • How does the area <objs> in <image> stand out uniquely from the rest?
NLVR	• Between the left image <image> and the right image <image>, could you tell me if the answer to <question> is True or False? • Reviewing both the left image <image> and the right image <image>, would you reckon <question> is True or False? • Given the left image <image> and the right image <image>, can you answer my query: <question>? Respond in True or False.
Spot Captioning	• Please provide a holistic description of the image <image> and output the position for each mentioned object in the format $[x_0, y_0, x_1, y_1]$ representing top-right and bottom-left corners of the bounding box. • Present a thorough insight into <image> and output every object's position using $[x_0, y_0, x_1, y_1]$, representing the bounding box's top-right and bottom-left corners.
CoSeg	• Find the common object in the input images <image>. There's only one common object. Display each object's mask in $[x_0, y_0, x_1, y_1, ..., x_{31}, y_{31}]$ format. Utilize <msep> to tell the masks apart. • Locate the common thing in the input images <image>. Only one common thing will be there. Present each thing's mask in $[x_0, y_0, x_1, y_1, ..., x_{31}, y_{31}]$ style. Use <msep> to differentiate the two masks.
AttCoSeg	• Find the two images which have a common object with matching attributes (shape, color, size, position), and segment it in both images. Show object mask in $[x_0, y_0, x_1, y_1, ..., x_{31}, y_{31}]$ style in both pictures. Make use of <msep> to tell apart the two masks. • Which input images have a mutual item with common attributes (shape, color, size, position)? Segment it in both images. Display object mask using $[x_0, y_0, x_1, y_1, ..., x_{31}, y_{31}]$ format in both images. Apply <msep> to differentiate the two masks.

Table E.1. **Examples of instructions** for different tasks used by VistaLLM to convert them into instruction-following format.



(a) **Performance of REC on RefCOCO with varying training samples.** We report the performance in terms of precision at IoU = 0.5, i.e., the prediction is deemed correct if its intersection over union (IoU) with the ground-truth box is larger than 0.5.

(b) **Performance of RES on RefCOCO with varying number of training samples.** We report the performance in terms of mIoU score.

Figure F.1. **Ablation on the number of training samples on the REC and RES task performance.** We start with only RES and REC datasets and gradually append datasets from other tasks using proper instructions. Increasing the number of samples helps produce better performance, showing the usefulness of an end-to-end, cohesive, and unified system where different tasks help improve each other.

train VistaLLM for two epochs in stage 1, setting all hyperparameters unchanged. In this setup, we observe a REC val score of 82.7%. Next, we add Visual Genome data to the training corpus, which results in a total of 1M samples, and re-train the model. Now, REC val accuracy increases to 84.0%. Similarly, appending PointQA data in the training corpus increases the performance by 1.3%, and appending LLaVa, Flickr, VQAv2, and COCO caption data yields a gain of another 0.7%. Finally, the 6.8M training corpus produces a final REC val accuracy of 88.1%. Hence, we observe that datasets from other image-level and region-level tasks help improve the performance of the REC task, which is the benefit of unified end-to-end training. We also see similar observations for the RES in Figure F.1b. Such a phenomenon also proves the scalability of our approach, which is important for large-scale unified training.

Image encoder: Next, we ablate different image encoders in Table F.1. We observe the best performance across most tasks with EVA[2] [31], while the CLIP-ViT-L/14-336px[3] follows closely. We use EVA-CLIP in our final model because the QFormer [16] pre-trained in InstructBLIP [9] uses EVA-CLIP, and it results in best compatibility with the instruction-guided image tokenizer module in our system.

Method	Cap.	RES Ref			VCR	iCoSeg	NLVR
	CIDEr	val	testA	testB	Q → AR	Av. $\mathcal{J}$	dev
VistaLLM-13B	**128.4**	**76.2**	**77.7**	**73.9**	**79.1**	**95.1**	**80.8**
w/ CLIP-ViT-L/14	127.9	75.5	76.3	72.1	79.3	94.7	80.2
w/ CLIP-ViT-L/14-336px	**128.4**	76.0	**77.7**	73.6	79.3	95.1	80.5
w/ CLIP-ViT-B/16	127.6	75.1	76.3	72.0	79.0	94.8	79.8

Table F.1. **Ablation with different image encoders.** By default, VistaLLM uses EVA-CLIP [31] pre-trained on LAION-400M [29]. We observe a small performance drop when using other image encoders.

G. Error Analysis

Although VistaLLM learns impressive reasoning and grounding capability across many different benchmarks, there are still some cases where the model fails to identify small and obscured objects, especially in cluttered environments. Figure G.1 shows seven such failure cases. In the RES example, the object "*teddy with arm up whose back in near brown plaid thing*" is hard to comprehend even for humans, and thus, VistaLLM can not identify the correct "*teddy*" the expression is referring to. In the REC example, the "*green hair tie*" is tiny and only visible when zoomed into the picture. VistaLLM fails to identify the girl who is wearing it. In the GREC example, in low-light conditions, the blue hoodie appears to be black, and VistaLLM wrongly outputs a bounding box, whereas the ground truth is no matching object. Similarly, in the NLVR2, GRES, and POPE examples, VistaLLM fails to recognize hindered and cluttered objects. We believe that more robust image features will alleviate such failure cases in the future. Moreover, similar to many LLMs, VistaLLM has the potential to generate harmful and unsafe outputs, which is also an active research topic.

H. Additional Qualitative Results

We provide additional qualitative results from VistaLLM-13B in Figures H.1, H.2, H.3, H.4, H.5, H.6, H.7, H.8, H.9, and H.10. Moreover, we illustrate multi-round conversational ability of VistaLLM in Figure H.11.

References

[1] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh.

[2] https://huggingface.co/QuanSun/EVA-CLIP/blob/main/EVA01_CLIP_g_14_psz14_s11B.pt

[3] https://huggingface.co/openai/clip-vit-large-patch14-336

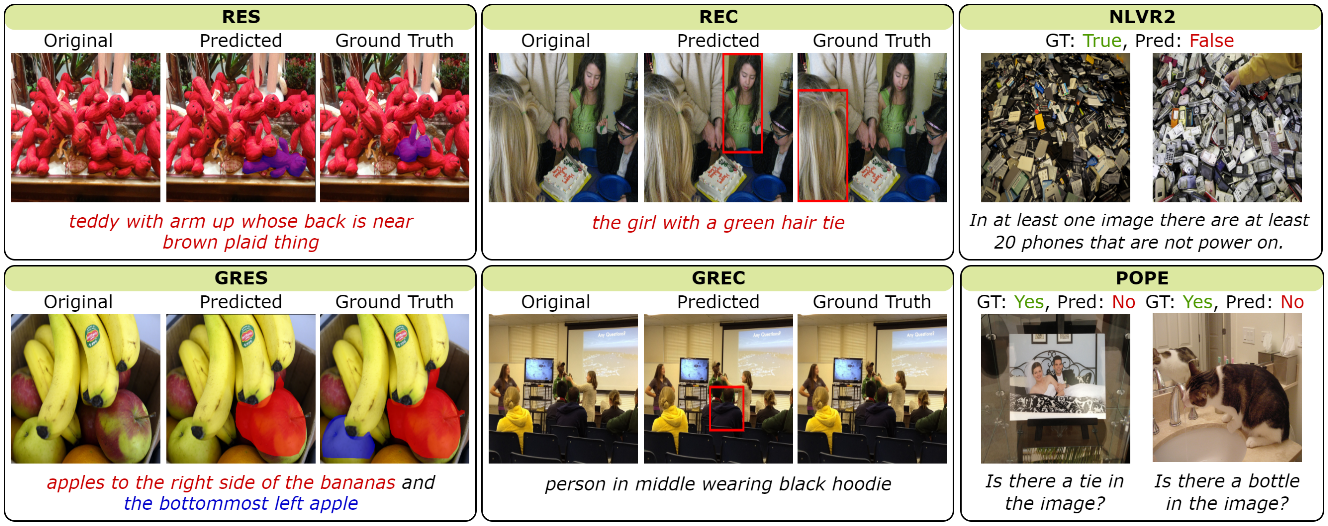
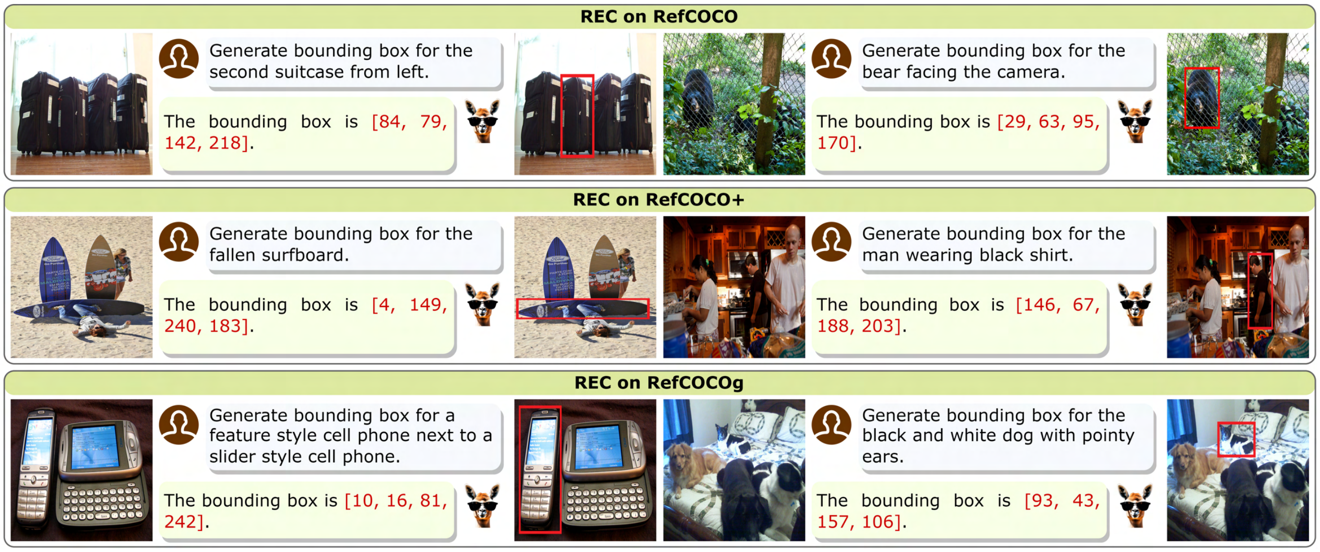

RES
Original
Predicted
Ground Truth
teddy with arm up whose back is near brown plaid thing
REC
Original
Predicted
Ground Truth
the girl with a green hair tie
NLVR2
GT: True, Pred: False
In at least one image there are at least 20 phones that are not power on.
GRES
Original
Predicted
Ground Truth
apples to the right side of the bananas and the bottommost left apple
GREC
Original
Predicted
Ground Truth
person in middle wearing black hoodie
POPE
GT: Yes, Pred: No
GT: Yes, Pred: No
Is there a tie in the image?
Is there a bottle in the image?

Figure G.1. **Limitations of our method:** Tiny and obscured objects, especially in cluttered and low-light environments, are hard to be accurately grounded. VistaLLM fails in such tough samples, which are even difficult to comprehend by humans.



Generate bounding box for the second suitcase from left.

The bounding box is [84, 79, 142, 218].

Generate bounding box for the bear facing the camera.

The bounding box is [29, 63, 95, 170].

Generate bounding box for the fallen surfboard.

The bounding box is [4, 149, 240, 183].

Generate bounding box for the man wearing black shirt.

The bounding box is [146, 67, 188, 203].

Generate bounding box for a feature style cell phone next to a slider style cell phone.

The bounding box is [10, 16, 81, 242].

Generate bounding box for the black and white dog with pointy ears.

The bounding box is [93, 43, 157, 106].

Figure H.1. **Referring Expression Comprehension (REC) on RefCOCO, RefCOCO+ and RefCOCOg by VistaLLM-13B.** REC aims to generate a bounding box around a single object described by a referring expression.

Vqa: Visual question answering. In *CVPR*, pages 2425–2433, 2015. 3

[2] Dhruv Batra, Adarsh Kowdle, Devi Parikh, Jiebo Luo, and Tsuhan Chen. icoseg: Interactive co-segmentation with intelligent scribble guidance. In *CVPR*, pages 3169–3176, 2010. 4

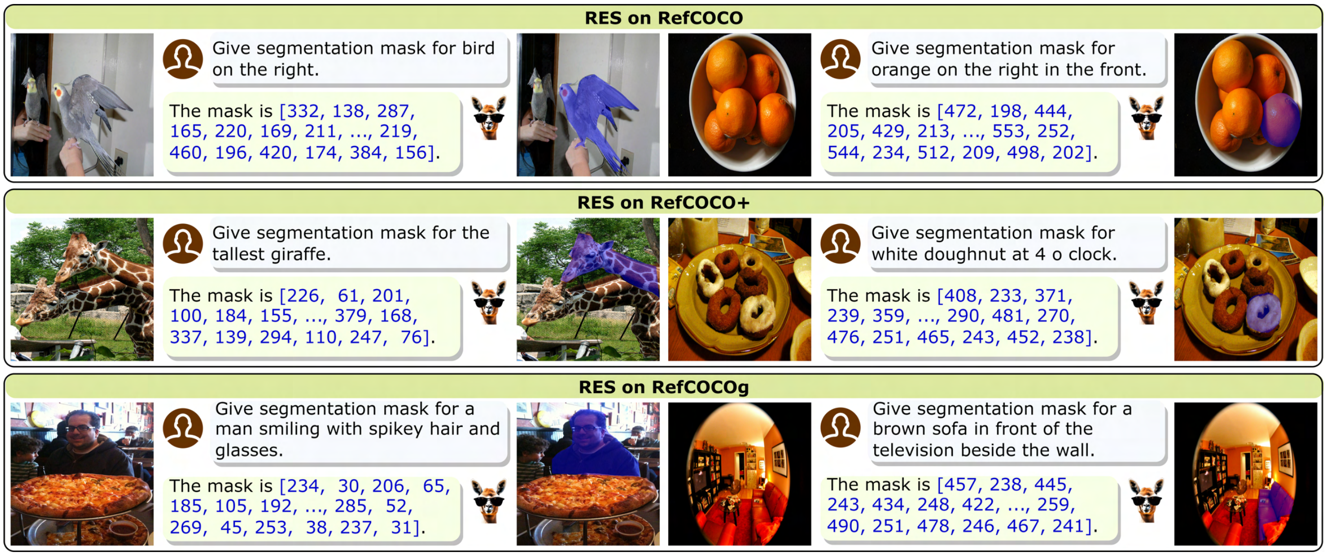
[3] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *NeurIPS*, 33:1877–1901, 2020. 4

[4] Chi Chen, Ruoyu Qin, Fuwen Luo, Xiaoyue Mi, Peng Li, Maosong Sun, and Yang Liu. Position-enhanced visual instruction tuning for multimodal large language models. *arXiv preprint arXiv:2308.13437*, 2023. 1, 2

[5] Jun Chen, Deyao Zhu, Xiaoqian Shen, Xiang Li, Zechun Liu, Pengchuan Zhang, Raghuraman Krishnamoorthi, Vikas Chandra, Yunyang Xiong, and Mohamed Elhoseiny. Minigpt-v2: large language model as a unified interface for vision-language multi-task learning. *arXiv preprint arXiv:2310.09478*, 2023. 1, 2

[6] Keqin Chen, Zhao Zhang, Weili Zeng, Richong Zhang, Feng Zhu, and Rui Zhao. Shikra: Unleashing multimodal llm's referential dialogue magic. *arXiv preprint arXiv:2306.15195*, 2023. 1, 2

[7] Ting Chen, Saurabh Saxena, Lala Li, David J Fleet, and Geoffrey Hinton. Pix2seq: A language modeling framework for



Give segmentation mask for bird on the right.

The mask is [332, 138, 287, 165, 220, 169, 211, ..., 219, 460, 196, 420, 174, 384, 156].

Give segmentation mask for orange on the right in the front.

The mask is [472, 198, 444, 205, 429, 213, ..., 553, 252, 544, 234, 512, 209, 498, 202].

Give segmentation mask for the tallest giraffe.

The mask is [226, 61, 201, 100, 184, 155, ..., 379, 168, 337, 139, 294, 110, 247, 76].

Give segmentation mask for white doughnut at 4 o clock.

The mask is [408, 233, 371, 239, 359, ..., 290, 481, 270, 476, 251, 465, 243, 452, 238].

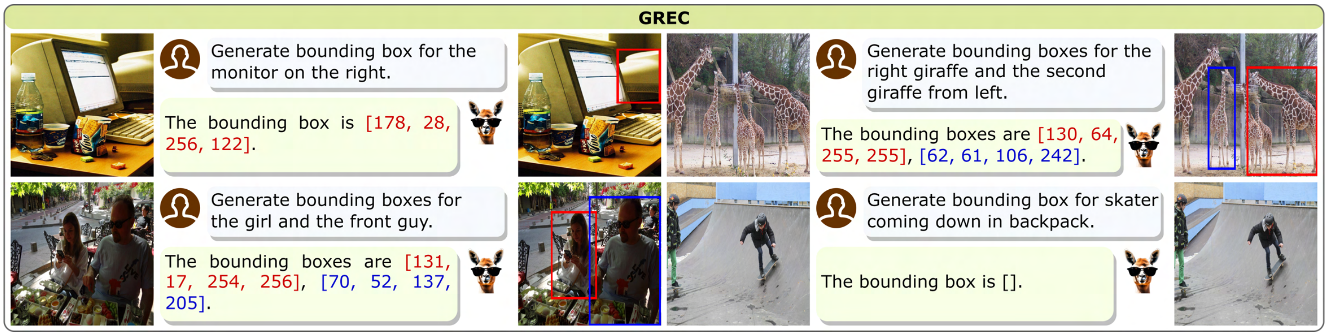
Give segmentation mask for a man smiling with spikey hair and glasses.

The mask is [234, 30, 206, 65, 185, 105, 192, ..., 285, 52, 269, 45, 253, 38, 237, 31].

Give segmentation mask for a brown sofa in front of the television beside the wall.

The mask is [457, 238, 445, 243, 434, 248, 422, ..., 259, 490, 251, 478, 246, 467, 241].

Figure H.2. **Referring Expression Segmentation (RES) on RefCOCO, RefCOCO+ and RefCOCOg by VistaLLM-13B.** RES aims to segment a single object described by a referring expression.



Generate bounding box for the monitor on the right.

The bounding box is [178, 28, 256, 122].

Generate bounding boxes for the right giraffe and the second giraffe from left.

The bounding boxes are [130, 64, 255, 255], [62, 61, 106, 242].

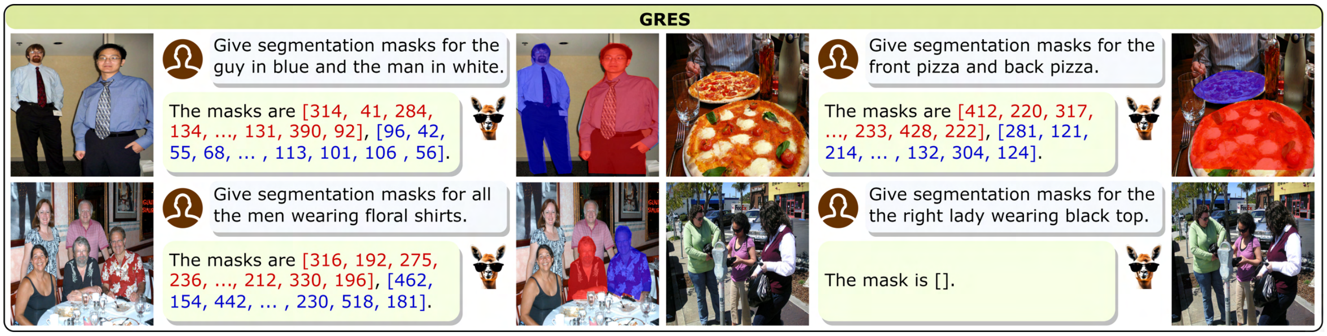
Generate bounding boxes for the girl and the front guy.

The bounding boxes are [131, 17, 254, 256], [70, 52, 137, 205].

Generate bounding box for skater coming down in backpack.

The bounding box is [].

Figure H.3. **Generalized Referring Expression Comprehension (GREC) on gRefCOCO by VistaLLM-13B.** GREC aims to identify all objects described by a referring expression and draw bounding boxes around every referred object. GREC also contains no-target expressions where the output is empty.



Give segmentation masks for the guy in blue and the man in white.

The masks are [314, 41, 284, 134, ..., 131, 390, 92], [96, 42, 55, 68, ... , 113, 101, 106 , 56].

Give segmentation masks for the front pizza and back pizza.

The masks are [412, 220, 317, ..., 233, 428, 222], [281, 121, 214, ... , 132, 304, 124].

Give segmentation masks for all the men wearing floral shirts.

The masks are [316, 192, 275, 236, ..., 212, 330, 196], [462, 154, 442, ... , 230, 518, 181].

Give segmentation masks for the the right lady wearing black top.

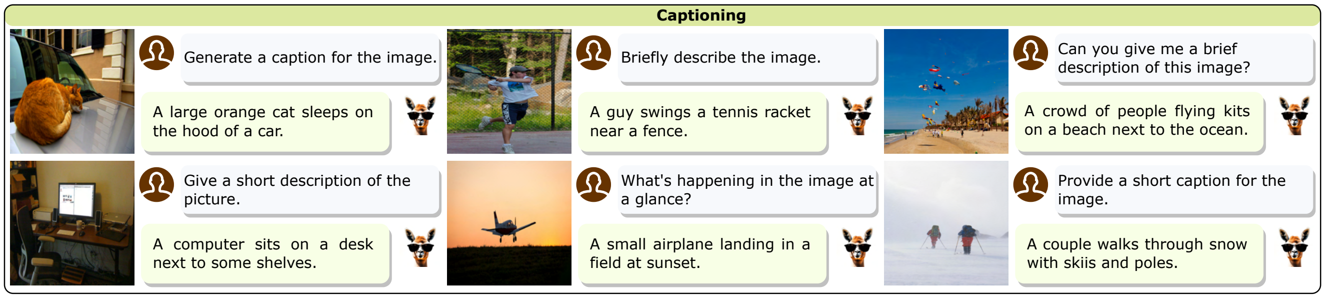
The mask is [].

Figure H.4. **Generalized Referring Expression Segmentation (GRES) on gRefCOCO by VistaLLM-13B.** GRES aims to identify all objects described by a referring expression and segment every referred object. GRES also contains no-target samples where the output is empty.

object detection. In *ICLR*, 2021. 3

[8] Ting Chen, Saurabh Saxena, Lala Li, Tsung-Yi Lin, David J Fleet, and Geoffrey E Hinton. A unified sequence interface for vision tasks. *NeurIPS*, 35:31333–31346, 2022. 3

[9] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale



Generate a caption for the image.

A large orange cat sleeps on the hood of a car.

Briefly describe the image.

A guy swings a tennis racket near a fence.

Can you give me a brief description of this image?

A crowd of people flying kits on a beach next to the ocean.

Give a short description of the picture.

A computer sits on a desk next to some shelves.

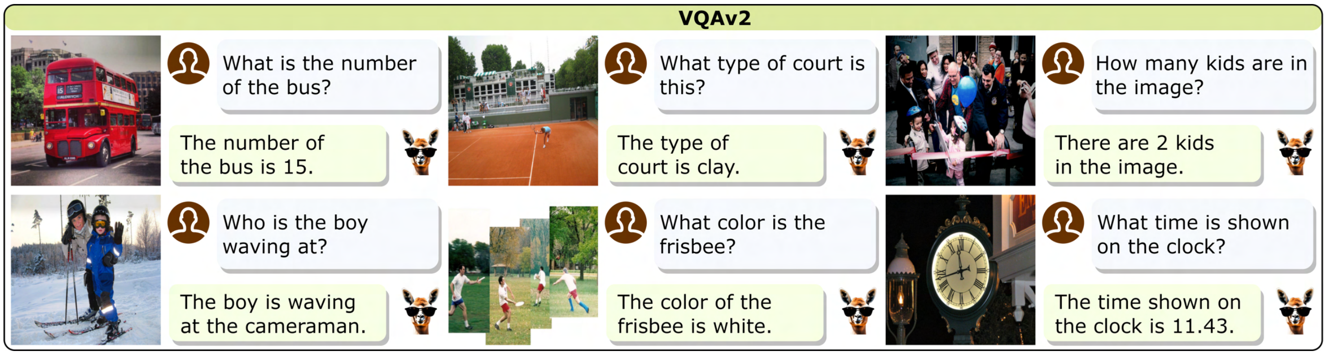
What's happening in the image at a glance?

A small airplane landing in a field at sunset.

Provide a short caption for the image.

A couple walks through snow with skiis and poles.

Figure H.5. **Image Captioning on COCO by VistaLLM-13B**, which aims to generate a short holistic description of the input image.



What is the number of the bus?

The number of the bus is 15.

What type of court is this?

The type of court is clay.

How many kids are in the image?

There are 2 kids in the image.

Who is the boy waving at?

The boy is waving at the cameraman.

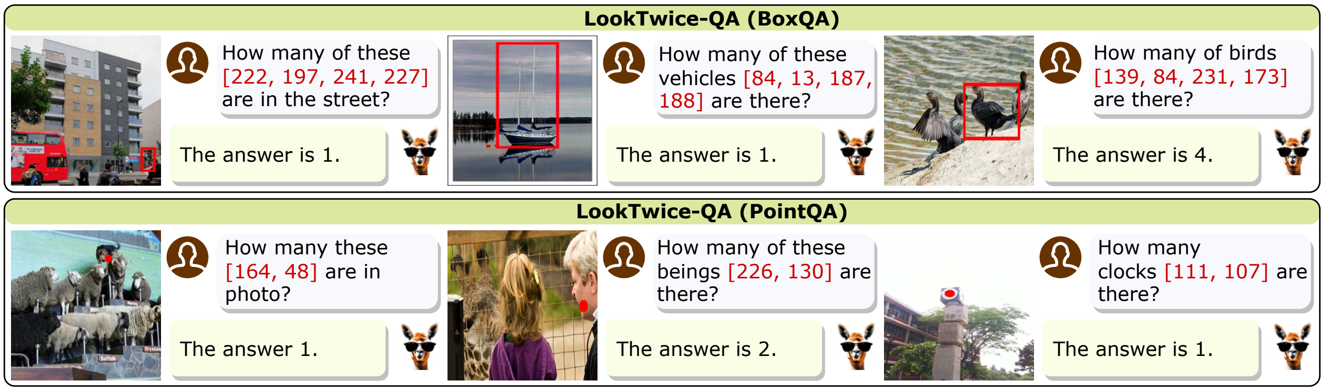
What color is the frisbee?

The color of the frisbee is white.

What time is shown on the clock?

The time shown on the clock is 11.43.

Figure H.6. **VQAv2 by VistaLLM-13B**, which aims to answer direct questions based on an input image.



How many of these [222, 197, 241, 227] are in the street?

The answer is 1.

How many of these vehicles [84, 13, 187, 188] are there?

The answer is 1.

How many of birds [139, 84, 231, 173] are there?

The answer is 4.

How many these [164, 48] are in photo?

The answer 1.

How many of these beings [226, 130] are there?

The answer is 2.

How many clocks [111, 107] are there?

The answer is 1.

Figure H.7. **Box Question Answering (BoxQA) and Point Question Answering (PointQA) on LookTwice-QA by VistaLLM-13B.** Given a question about a specified region in the image, either mentioning a point or a box, this task needs to comprehend the area in the context of the whole image to produce the correct answer.

Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning. In *NeurIPS*, 2023. 2, 6

[10] Alon Faktor and Michal Irani. Co-segmentation by composition. In *ICCV*, pages 1297–1304, 2013. 4

[11] Shuting He, Henghui Ding, Chang Liu, and Xudong Jiang. Grec: Generalized referring expression comprehension. *arXiv preprint arXiv:2308.16182*, 2023. 3

[12] Dongsheng Jiang, Yuchen Liu, Songlin Liu, Xiaopeng Zhang, Jin Li, Hongkai Xiong, and Qi Tian. From clip to dino: Visual encoders shout in multi-modal large language models. *arXiv preprint arXiv:2310.08825*, 2023. 1, 2

[13] Douwe Kiela, Hamed Firooz, Aravind Mohan, Vedanuj Goswami, Amanpreet Singh, Pratik Ringshia, and Davide Testuggine. The hateful memes challenge: Detecting hate speech in multimodal memes. In *NeurIPS*, pages 2611–2624, 2020. 4

[14] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *IJCV*, 123:32–73, 2017. 3

[15] Xin Lai, Zhuotao Tian, Yukang Chen, Yanwei Li, Yuhui Yuan, Shu Liu, and Jiaya Jia. Lisa: Reasoning segmentation

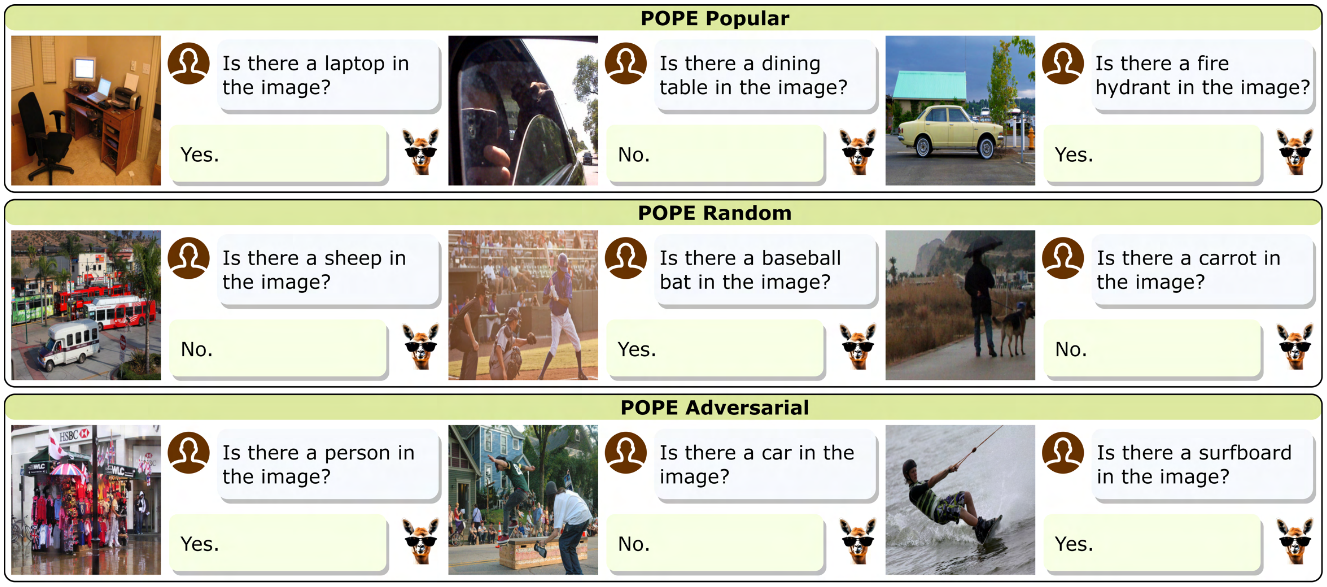

POPE Popular
Is there a laptop in the image?
Yes.
Is there a dining table in the image?
No.
Is there a fire hydrant in the image?
Yes.
POPE Random
Is there a sheep in the image?
No.
Is there a baseball bat in the image?
Yes.
Is there a carrot in the image?
No.
POPE Adversarial
Is there a person in the image?
Yes.
Is there a car in the image?
No.
Is there a surfboard in the image?
Yes.

Figure H.8. **Object Hallucination Evaluation of VistaLLM-13B on POPE benchmark.** The task aims to input a query inquiring about the existence of an object, and the model is expected to generate a response in the form of either "yes/no."

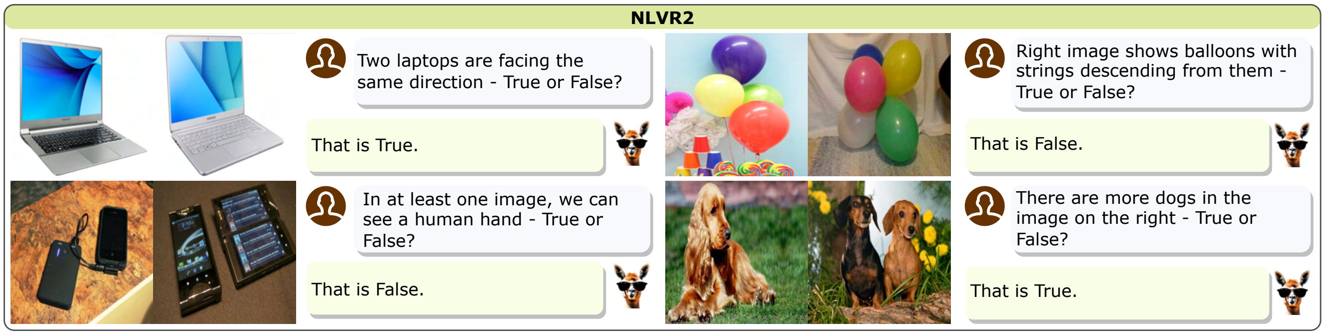

NLVR2
Two laptops are facing the same direction - True or False?
That is True.
Right image shows balloons with strings descending from them - True or False?
That is False.
In at least one image, we can see a human hand - True or False?
That is False.
There are more dogs in the image on the right - True or False?
That is True.

Figure H.9. **Natural Language for Visual Reasoning (NLVR2) by VistaLLM-13B.** Given a pair of input images and a question, the model must reason both images to produce the answer correctly.

via large language model. *arXiv preprint arXiv:2308.00692*, 2023. 1, 2

[16] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. BLIP-2: bootstrapping language-image pre-training with frozen image encoders and large language models. In *ICML*, 2023. 6

[17] Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. In *EMNLP*, 2023. 4

[18] 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 *ECCV*, pages 740–755, 2014. 3

[19] Chang Liu, Henghui Ding, and Xudong Jiang. Gres: Generalized referring expression segmentation. In *CVPR*, pages 23592–23601, 2023. 3

[20] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023. 2, 4

[21] Jingyu Liu, Liang Wang, and Ming-Hsuan Yang. Referring expression generation and comprehension via attributes. In *ICCV*, pages 4856–4864, 2017. 3

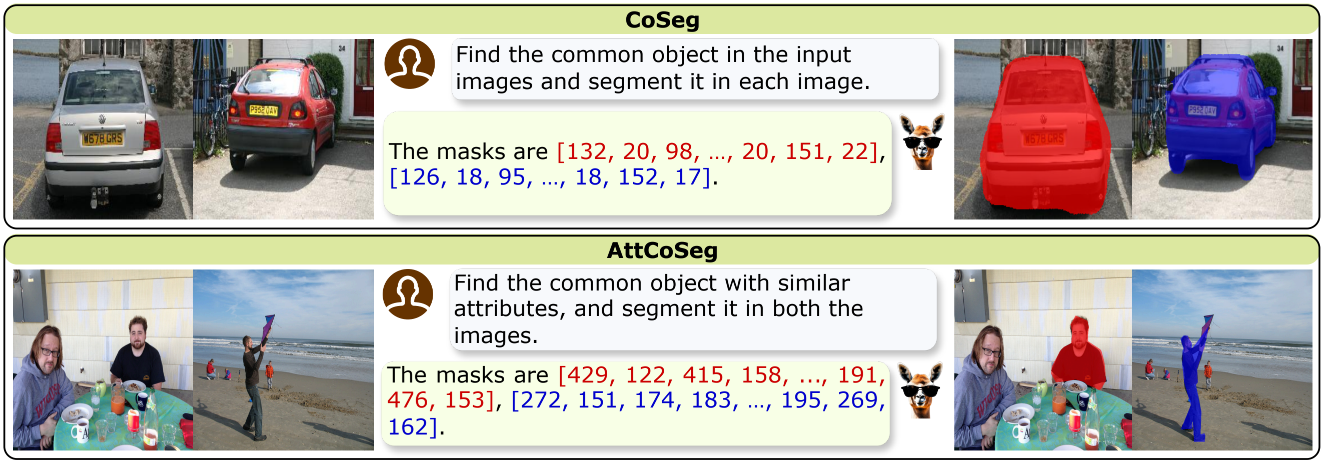
[22] Jiang Liu, Hui Ding, Zhaowei Cai, Yuting Zhang, Ravi Kumar Satzoda, Vijay Mahadevan, and R Manmatha. Polyformer: Referring image segmentation as sequential polygon generation. In *CVPR*, pages 18653–18663, 2023. 3

[23] Pan Lu, Liang Qiu, Jiaqi Chen, Tony Xia, Yizhou Zhao, Wei Zhang, Zhou Yu, Xiaodan Liang, and Song-Chun Zhu. Iconqa: A new benchmark for abstract diagram understanding and visual language reasoning. In *NeurIPS Datasets and Benchmarks Track*, 2021. 4

[24] Arjun Mani, Nobline Yoo, Will Hinthorn, and Olga Russakovsky. Point and ask: Incorporating pointing into visual question answering. *arXiv preprint arXiv:2011.13681*, 2020. 4

[25] Varun K Nagaraja, Vlad I Morariu, and Larry S Davis. Modeling context between objects for referring expression understanding. In *ECCV*, pages 792–807. Springer, 2016. 3

[26] Zhiliang Peng, Wenhui Wang, Li Dong, Yaru Hao, Shaohan



CoSeg

Find the common object in the input images and segment it in each image.

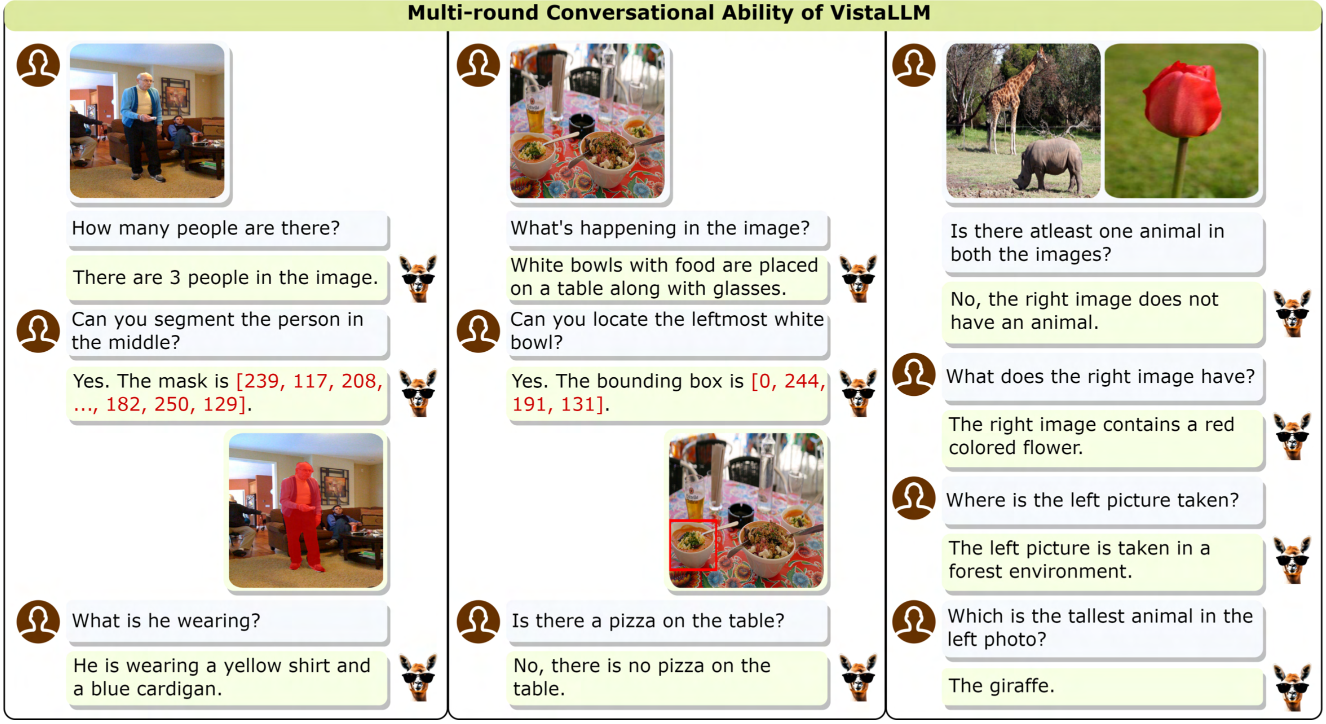
The masks are [132, 20, 98, ..., 20, 151, 22], [126, 18, 95, ..., 18, 152, 17].

AttCoSeg

Find the common object with similar attributes, and segment it in both the images.

The masks are [429, 122, 415, 158, ..., 191, 476, 153], [272, 151, 174, 183, ..., 195, 269, 162].

Figure H.10. **CoSeg and AttCoSeg by VistaLLM-13B.** Given a set of input images, CoSeg aims to find and segment a common object in every image. AttCoSeg is the more challenging scenario where the input images contains a common object with similar attributes. VistaLLM is expected to segment the object in both images.



Multi-round Conversational Ability of VistaLLM

How many people are there?

There are 3 people in the image.

Can you segment the person in the middle?

Yes. The mask is [239, 117, 208, ..., 182, 250, 129].

What is he wearing?

He is wearing a yellow shirt and a blue cardigan.

What's happening in the image?

White bowls with food are placed on a table along with glasses.

Can you locate the leftmost white bowl?

Yes. The bounding box is [0, 244, 191, 131].

Is there a pizza on the table?

No, there is no pizza on the table.

Is there atleast one animal in both the images?

No, the right image does not have an animal.

What does the right image have?

The right image contains a red colored flower.

Where is the left picture taken?

The left picture is taken in a forest environment.

Which is the tallest animal in the left photo?

The giraffe.

Figure H.11. **Multi-round Conversational Ability of VistaLLM-13B.** The images are taken from COCO. VistaLLM can address all possible grounding and reasoning tasks across single and multiple input images.

Huang, Shuming Ma, and Furu Wei. Kosmos-2: Grounding multimodal large language models to the world. *arXiv preprint arXiv:2306.14824*, 2023. 1, 2

[27] Renjie Pi, Jiahui Gao, Shizhe Diao, Rui Pan, Hanze Dong, Jipeng Zhang, Lewei Yao, Jianhua Han, Hang Xu, and Lingpeng Kong Tong Zhang. Detgpt: Detect what you need via reasoning. *arXiv preprint arXiv:2305.14167*, 2023. 1, 2

[28] Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *CVPR*, pages 2641–2649, 2015. 3

[29] Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *arXiv preprint arXiv:2111.02114*, 2021. 6

[30] Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang,

Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *CVPR*, pages 8317–8326, 2019. 4

[31] Quan Sun, Yuxin Fang, Ledell Wu, Xinlong Wang, and Yue Cao. Eva-clip: Improved training techniques for clip at scale. *arXiv preprint arXiv:2303.15389*, 2023. 6

[32] 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. In *NeurIPS*, 2023. 1, 2

[33] Weihan Wang, Qingsong Lv, Wenmeng Yu, Wenyi Hong, Ji Qi, Yan Wang, Junhui Ji, Zhuoyi Yang, Lei Zhao, Xixuan Song, et al. Cogvlm: Visual expert for pretrained language models. *arXiv preprint arXiv:2311.03079*, 2023. 1, 2

[34] John Winn, Antonio Criminisi, and Thomas Minka. Object categorization by learned universal visual dictionary. In *ICCV*, pages 1800–1807, 2005. 4

[35] Chenfei Wu, Shengming Yin, Weizhen Qi, Xiaodong Wang, Zecheng Tang, and Nan Duan. Visual chatgpt: Talking, drawing and editing with visual foundation models. *arXiv preprint arXiv:2303.04671*, 2023. 1, 2

[36] Yixuan Wu, Zhao Zhang, Chi Xie, Feng Zhu, and Rui Zhao. Advancing referring expression segmentation beyond single image. In *ICCV*, pages 2628–2638, 2023. 4

[37] 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. In *ICLR*, 2024. 1, 2

[38] Licheng Yu, Patrick Poirson, Shan Yang, Alexander C Berg, and Tamara L Berg. Modeling context in referring expressions. In *ECCV*. Springer, 2016. 3

[39] Rowan Zellers, Yonatan Bisk, Ali Farhadi, and Yejin Choi. From recognition to cognition: Visual commonsense reasoning. In *CVPR*, pages 6720–6731, 2019. 3

[40] 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. 1, 2

[41] 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. 1, 2

[42] Chaoyang Zhu, Yiyi Zhou, Yunhang Shen, Gen Luo, Xingjia Pan, Mingbao Lin, Chao Chen, Liujuan Cao, Xiaoshuai Sun, and Rongrong Ji. Seqtr: A simple yet universal network for visual grounding. In *ECCV*, pages 598–615. Springer, 2022. 3

[43] Yuke Zhu, Oliver Groth, Michael Bernstein, and Li Fei-Fei. Visual7w: Grounded question answering in images. In *CVPR*, pages 4995–5004, 2016. 4