A summary is provided on page 10 . It is recommended to go through it before reading the full paper

VCoder: Versatile Vision Encoders for Multimodal Large Language Models

Jitesh Jain[1] Jianwei Yang[2] Humphrey Shi[1,3]
[1]SHI Labs @ Georgia Tech [2]Microsoft Research [3]Picsart AI Research (PAIR)

https://github.com/SHI-Labs/VCoder

Abstract

*Humans possess the remarkable skill of Visual Perception, the ability to see and understand the seen, helping them make sense of the visual world and, in turn, reason. Multimodal Large Language Models (MLLM) have recently achieved impressive performance on vision-language tasks ranging from visual question-answering and image captioning to visual reasoning and image generation. However, when prompted to identify or count (perceive) the entities in a given image, existing MLLM systems fail. Working towards developing an accurate MLLM system for perception and reasoning, we propose using Versatile vision enCoders (**VCoder**) as perception eyes for Multimodal LLMs. We feed the VCoder with perception modalities such as segmentation or depth maps, improving the MLLM's perception abilities. Secondly, we leverage the images from COCO and outputs from off-the-shelf vision perception models to create our COCO Segmentation Text (**COST**) dataset for training and evaluating MLLMs on the object perception task. Thirdly, we introduce metrics to assess the object perception abilities in MLLMs on our COST dataset. Lastly, we provide extensive experimental evidence proving the VCoder's improved object-level perception skills over existing Multimodal LLMs, including GPT-4V. We open-source our dataset, code, and models to promote research.*

1. Introduction

> *'Perception is the soil; reasoning, the seed. Without fertile ground, the seed cannot flourish.'*
> (GPT-4 [51], 2023)

The ability to think and reason is one of the most remarkable traits that help humans function daily. Generally, understanding the environment precedes the act of thinking and reasoning [30]. Following the success of ChatGPT-like instruction following AI agents [3, 11, 50, 51, 64] at language understanding and reasoning, researchers have leveraged LLMs to develop instruct frameworks [14, 42, 72, 78] that can understand vision and language inputs in an effort to imitate human perception and reasoning ability. We refer to such systems as Multimodal LLMs (MLLM). Although MLLMs exhibit the ability to perform complex vision-language tasks like visual captioning [2, 4, 41], image generation [27, 29, 61], visual reasoning and grounding [23, 54, 76], they often display sub-par performance at simple tasks like counting objects (Fig. 1). As shown in Fig. 2, MLLMs output incorrect object counts (*people, chairs*) and hallucinate about the presence (*handbag, couch*) of certain objects when prompted to identify entities in a visual input. The perception performance is much worse

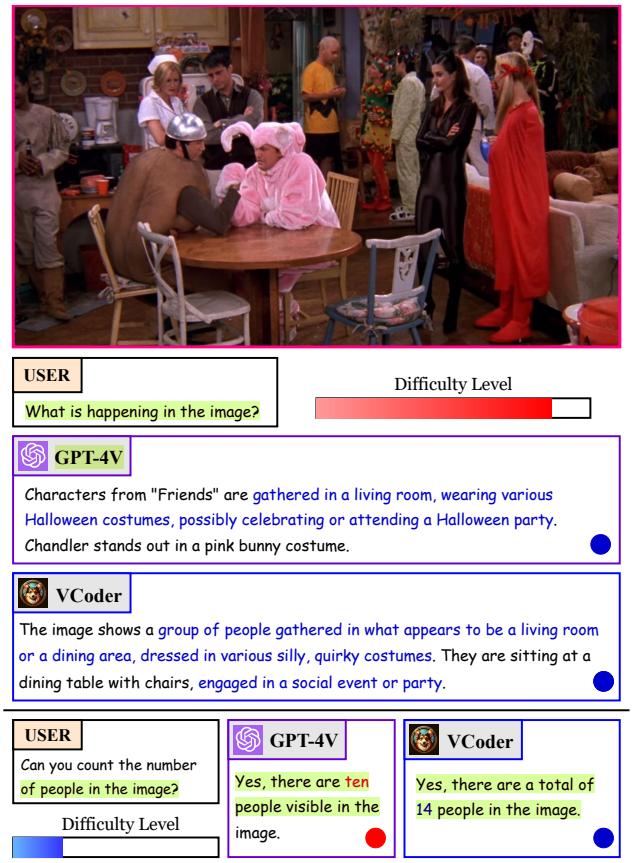


Figure 1. GPT-4V [51] (access date: Dec 16, 2023) returns impressive responses when prompted to describe complex visual scenes. However, it fails at the simple task of counting in the same scene. Our VCoder returns the correct count of people.

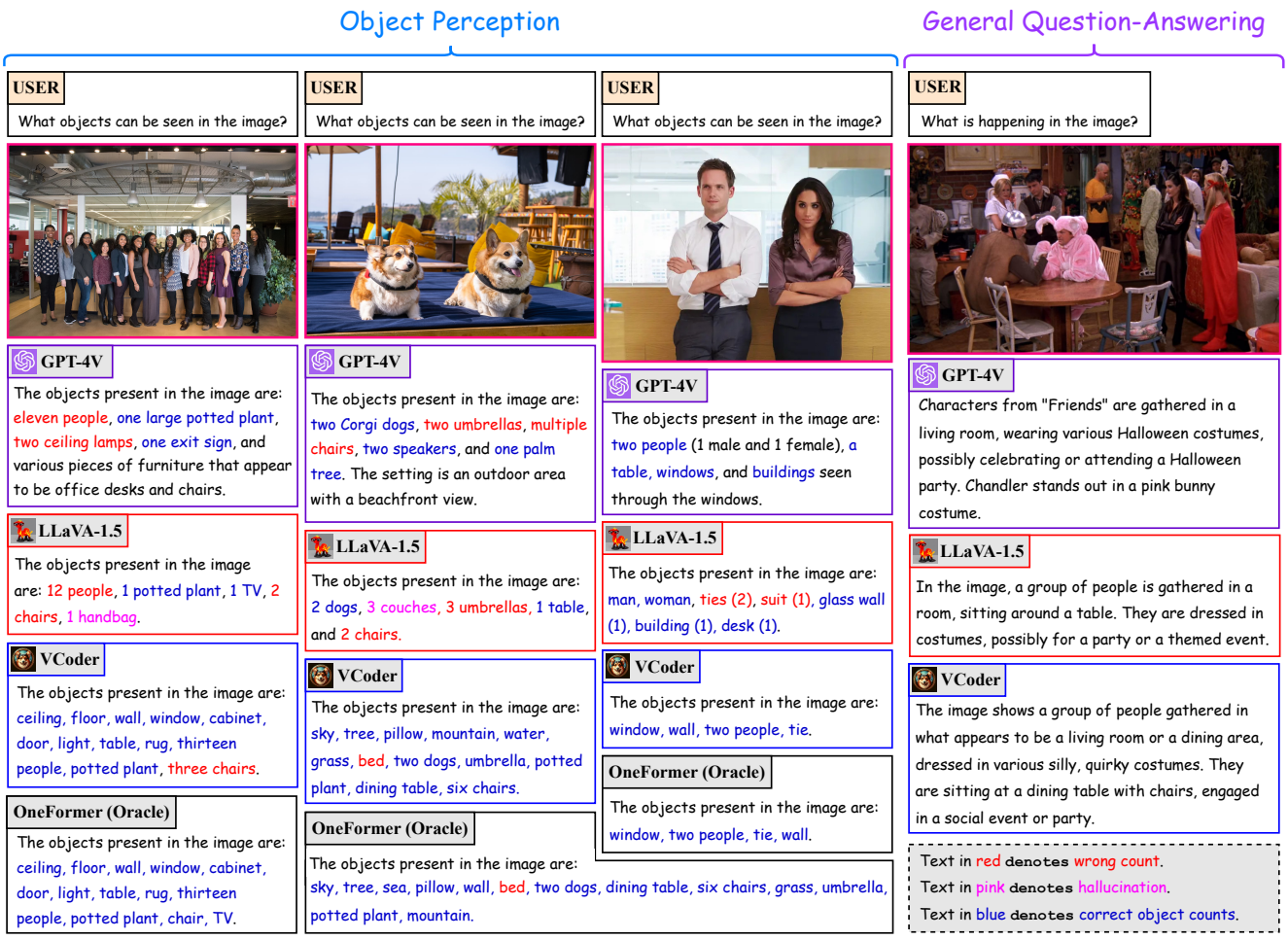


Figure 2. **MLLMs counting and identifying objects.** As shown in the first column, GPT-4V [51] (access date: Dec 16, 2023) and LLaVA-1.5 [41] both fail at counting *people*. Moreover, LLaVA-1.5 [41] misses background entities like *window, wall, etc.* and hallucinates about the presence of a handbag. VCoder can predict the *people* counts and other background entities accurately except *chairs*. Similarly, in the second column, GPT-4V and LLaVA-1.5 fail at counting *chairs* while the VCoder matches the Oracle's performance. Notably, all MLLMs can perceive objects accurately for a non-cluttered image in the third column, with LLaVA-1.5 failing at counting ties. Our VCoder can also accurately perform general question-answering tasks, as shown in the fourth column. We treat OneFormer [26] as the Oracle for object perception. Red text represents counting mistakes; pink text represents hallucination; blue text represents correct object perception.

when the scenes are cluttered with many entities. Consequently, a natural question arises: *"How to develop MLLM systems that respond to **perception** questions accurately?"*

This work aims to improve Multimodal LLMs at the simple yet fundamental object-level perception skills, including counting. Our motivation stems from the intuition that one can only describe and reason about a visual scene with the correct understanding of the entities in the image. In our effort to develop an accurate Multimodal LLM perception system, we face three significant challenges: (i) the scarcity of a vision-language dataset focused on the object perception task; (ii) existing open-sourced Multimodal LLMs usually use the ViT from CLIP [55] with an RGB image as input as the visual component that majorly focuses only on salient objects, and (iii) the absence of evaluation metrics to quantitatively measure Multimodal LLMs' object perception and in particular, counting skills. We list our efforts to overcome the issues above in the following paragraphs.

The contemporary vision-language models [14, 36, 55] owe their success to the availability of large-scale image-text datasets [7, 53, 60]. However, these datasets are more focused on image captioning [35] and VQA [1] tasks, making them unfit for training Multimodal LLMs for basic perception skills like object identification and counting. To overcome the scarcity of fundamental perception-focused image-text data, we leverage images from the COCO [38] dataset and use predictions from off-the-shelf visual perception models [26, 52, 57] to prepare a **CO**CO **S**egmentation **T**ext (**COST**) dataset comprising of question-answer pairs about the objects (background and foreground) present in

Nicely done!

What are the problems that authors are trying to address?
What are the challenges in building a more robust solution?

each image. We provide more details in Sec. 3.1.

Inspired by diffusion models that add various perception "control" or "context" images [48, 70, 71, 74] as auxiliary inputs to aid image generation, we propose feeding extra perception modalities as control inputs through additional vision encoders, which we term as our Versatile vision enCoders (**VCoder**). In this work, we focus on the task of object perception and leverage a segmentation map, depth map, or both as the control inputs; however, the same design can be extended to other modalities. Our VCoder projects the control inputs' information into the LLM's space as shown in Fig. 4. We hypothesize that this added control helps the MLLM improve its object perception ability.

Lastly, owing to the absence of metrics to quantify the counting ability in MLLMs, we propose computing a count score (**CS**) using one-to-one matching of object words in the ground truth and MLLM's answer. We also compute a hallucination score (**HS**) based on the extra objects in the MLLM's response that are absent from the ground truth. Similarly, we introduce a depth score (**DS**) to quantify the object order prediction performance in MLLMs.

Among the open-source MLLMs, we choose LLaVA-1.5 [41] as our base MLLM due to its impressive performance. Our extensive experimental analysis demonstrates the importance of our COST dataset and VCoder LLaVA-1.5's improved perception ability. To summarize, our contributions are as follows:

- We propose using extra (perception) control inputs and feeding those to a **V**ersatile en**Coder** (**VCoder**) for improved object perception performance.
- We introduce a COCO Segmentation Text (**COST**) dataset to train and evaluate Multimodal LLM systems on the fundamental object-level perception tasks of object identification, counting, and order prediction.
- Furthermore, to quantify the object perception ability in MLLMs, we propose calculating a count score (**CS**), a hallucination score (**HS**) and a depth score (**DS**). Our experiments show that the VCoder-adapted LLaVA-1.5 outperforms the baseline MLLMs on all metrics when validated on the COST dataset.

2. Related Work

2.1. Visual Perception

The fundamental nature of visual perception makes it a critical component in MLLM systems. The task of perception can be divided into sub-tasks, including dense prediction tasks like image segmentation [25, 26, 44, 66] and depth estimation [16, 18, 57], and sparse prediction tasks like object detection [6, 67] and pose estimation [13, 62]. In the deep learning era, initial methods tackled the perception task using CNN based methods [8, 9, 22, 31, 32, 58, 62] with recent methods shifting to the use of vision transformer based architectures [10, 18, 26, 57, 69, 79]. In this work, we tackle the fundamental task of object-level perception, mainly focusing on predicting names, counts, and order of objects in an image using MLLMs.

2.2. Visual Understanding with LLMs

Using LLMs for vision applications is not a new concept. In a nutshell, developing Multimodal LLMs involves projecting [2, 4, 36, 65] the features from a vision encoder [15, 56] to the embedding space of a language model (LLM) [11, 63, 64], and, instruction-tuning on a vision-language dialog dataset.

LLaVA [42] proposed a pipeline to convert existing image-text data into dialog format and then finetuned a CLIP [55] and LLaMA [63] model end-to-end on their collected dataset showing one of the earliest evidence of visual-language instruction tuning. Concurrent to LLaVA, MiniGPT-4 [78] used the visual encoder from BLIP2 [36] and used a linear layer for projecting visual features into Vicuna's [11] feature space. InstructBLIP [14] open-sourced a collection of 16 different datasets covering various vision tasks like VQA, reasoning, captioning, classification, etc., and finetuned a BLIP2 model on their dataset. mPLUG-Owl [72] proposed using a vision abstractor and finetuning the vision encoder. More recently, LLaVA-1.5 [41] proposed using an MLP as the projector and finetuned on academic instruction datasets to achieve state-of-the-art performance on various benchmarks [12, 17, 24, 37]. Among various open-source MLLMs [5, 23, 33, 34, 73], we chose LLaVA-1.5 as our baseline due to its superior performance.

2.3. Perception Hallucination in MLLMs

Since the introduction of LLMs, there has been a comprehensive study about their ability to hallucinate [75] in the NLP community. However, the phenomenon of hallucination in Multimodal LLMs has received comparatively less attention. LRV-Instruction [40] introduced a new instruction-tuning dataset containing 400k visual instructions to prevent hallucination in MLLMs and measured performance treating responses from GPT-4 [51] as ground truths. More recently, HallusionBench [39] quantitatively benchmarked various failure modes in MLLMs that lead to hallucinations based primarily on logical consistency and reasoning. Unlike these works that tried to benchmark MLLMs mainly on VQA-type tasks, this paper focuses on the object-level hallucination in MLLMs.

The two closest works to our objective are POPE [37] and CHAIR [59]. On the one hand, POPE [37] tried to measure hallucination in MLLMs using a binary "Yes"-"No" answer policy in response to questions based on the absence or presence of an object in the image. On the other hand, CHAIR [59] focused on measuring hallucination in image captioning based on only words and not counts for

Problems with existing metrics like POPE and CHAIR

Figure 3. **Organization of the COST dataset**. We incorporate the images from COCO [38], the questions from GPT-4 [51], and the segmentation outputs from OneFormer [26] in a question-answer format for training and evaluating MLLMs on the object identification task. We also extend COST to the object order perception task by incorporating depth map outputs from DINOv2 [52] DPT [57]. COST can be extended to more object-level tasks by similarly incorporating other modalities (for example, keypoint maps).

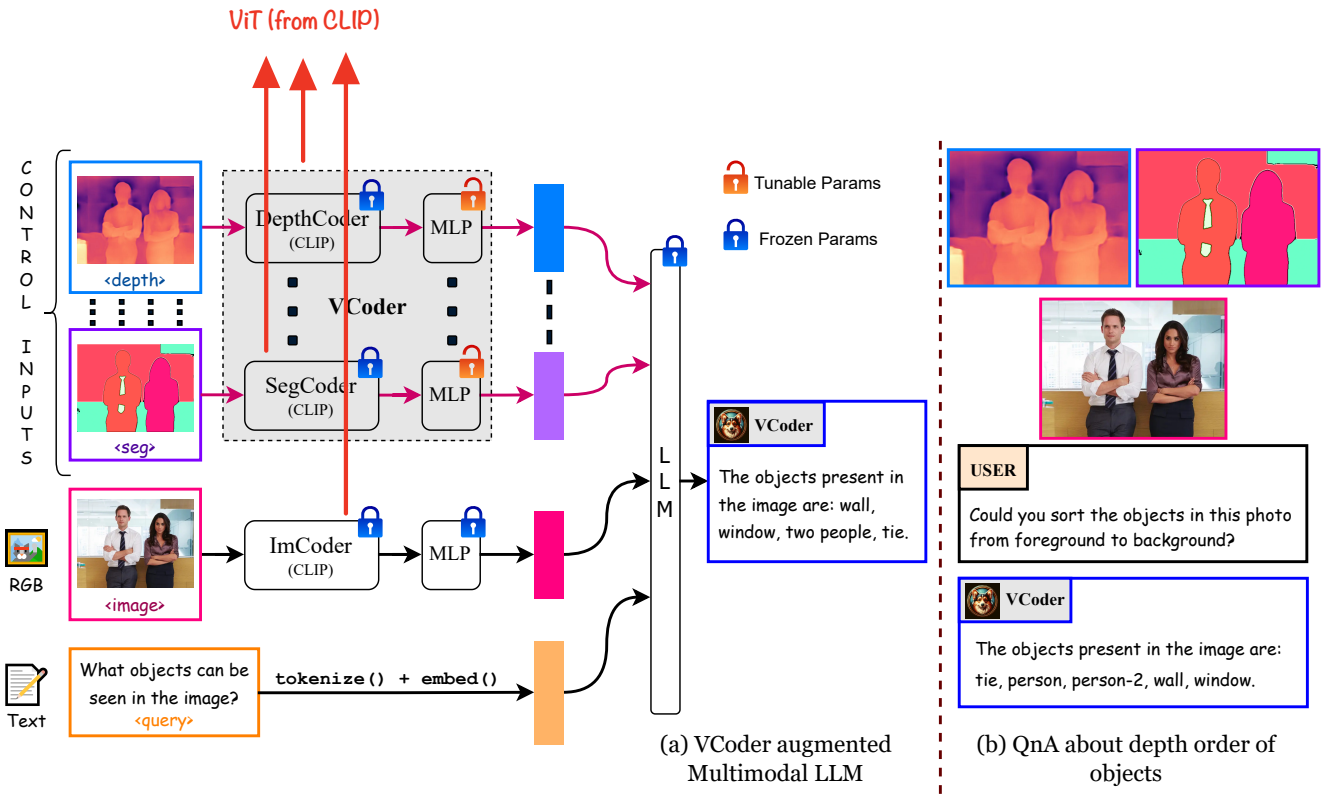
the objects. In our work, we consider not only object words but also the corresponding count to compute an object-level count score and hallucination score.

3. Object Identification with MLLMs

Suppose you are invited to a Halloween party and want to bring candies for every person at that party. You ask your friend to send you a picture (Fig. 1) of the party room so that you can estimate the number of people and the number of candies you need to buy. In a hurry, you ask GPT-4V [51]: "*Can you count the number of people in the image?*", and it responds: "*Yes, there are ten people visible in the image.*". Excited, you arrive at the party with ten candies but wait, you see fourteen people! Confused, you look at the image your friend sent you, and you can count fourteen people in that image, realizing that GPT-4V fails at the simple task of counting the people in the picture. At the same time, it can accurately describe the happening of a Halloween party in the image (Fig. 1). We refer to the phenomenon of Multi-modal LLMs failing at simple visual perception tasks while succeeding at complex visual reasoning tasks as Moravec's Paradox [47] in perception.

We hypothesize that one of the main reasons for the above phenomenon is the absence of conversations covering object identification for not only the salient objects but also the objects in the background from the instruction-tuning data for MLLMs. To overcome this issue, we prepare the COCO Segmentation Text (**COST**) dataset with COCO [38] images and create sentences using the output from an image segmentation model [26] to obtain an image-text dataset to train and evaluate MLLMs for object perception MLLMs. Moreover, we also introduce a segmentation map as a control image input to the MLLM for better performance and quantify object perception performance with a count score (**CS**) and a hallucination score (**HS**).

Nah, not convinced!



(a) VCoder augmented Multimodal LLM

(b) QnA about depth order of objects

Figure 4. **Adapting Multimodal LLMs for accurate object perception with VCoder.** **(a)** We add our VCoder as an adapter to the LLaVA-1.5 [41] and feed perception modalities as extra control inputs for improved object perception performance. During training, we freeze the components from LLaVA-1.5 (ImCoder, MLP, and LLM) to retain the original reasoning performance. **(b)** Using depth map and segmentation map as the control inputs to VCoder for the object order perception task.

3.1. COST to Identify Objects with MLLMs

We find that image segmentation methods [10, 26] can accurately identify salient (foreground objects like *people, cars, etc.*) and background objects (like *sky, wall, etc.*) in a given scene. Guided by this finding, we use images from the COCO [38] dataset and obtain the corresponding segmentation outputs from OneFormer [26], a state-of-the-art image segmentation model. Next, we extract the object (class) names and counts from the segmentation outputs and convert them into a sentence form for the ground-truth answer: "*The objects present in the image are: [CNT_1] [OBJ_1], [CNT_2] [OBJ_2], ..., [CNT_N] [OBJ_N].*", with *[OBJ_i]* representing the object name and *[CNT_i]* representing the count (if greater than one) for the i^{th} object in the image. We prompt GPT-4 [51] to collect a bucket of questions for three different object identification tasks: semantic, instance, and panoptic, corresponding to the three different image segmentation tasks. Finally, as shown in Fig. 3, we organize the images from COCO, segmentation maps from OneFormer, questions from GPT-4, and sentences containing object information into a question-answer format to construct our **CO**CO **S**egmentation **T**ext (**COST**) dataset for training and evaluating MLLMs on the object identification task.

Statistically, we prompt GPT-4 [51] to return 20 questions for each question bucket (panoptic, semantic, and instance). In total, we used 280k images from the `train2017`, `test2017`, and `unlabeled2017` splits of the COCO [38] dataset and corresponding segmentation outputs from OneFormer [26] to form the visual component of the COST training dataset. Similarly, we prepare a COST validation split using the 5k images from the `val2017` split of the COCO dataset.

Note that a similar approach can extend the COST dataset to other perception modalities. In this work, we incorporate the depth map modality into our COST dataset for the object order perception task. Particularly, we leverage the publicly available DINOv2 [52] DPT [57] model to obtain depth maps for COCO images and use the panoptic mask (from OneFormer [26]) to estimate the depth order of objects in an image. We format the obtained ordering of objects into the text with the template: *"The depth order for objects present in the image is: [OBJ_1], [OBJ_2], ..., [OBJ_J]."*, with *[OBJ_j]* representing the j^{th} object name. To maintain relative ordering among objects belonging to the same class, we append a count number to the second and later objects, as shown in the bottom right of Fig. 3 for *person* and *person-2*. Similar to the previous setting, we prompt GPT-4 [51] to return 20 questions for the object order perception task. We provide a detailed flow of obtaining ground-truth object orders in the appendix.

3.2. VCoder for Multimodal LLMs

We notice that existing open-source Multimodal LLMs generally use the ViT [15] from CLIP [56] as the image encoder

This is a weak hypothesis IMO

(ImCoder) during instruction tuning. We reason that the ViT focuses mainly on salient objects because it is trained against captions, which leave out information about background regions. We argue that identifying objects in the background is critical for a Multimodal LLM to become skilled at perception. To overcome this limitation, we introduce a segmentation map as a control input [48, 74] into our Multimodal LLM. Specifically, we use the segmentation map from OneFormer [26] and project it to the LLM's embedding space using a pretrained ViT [15] (from CLIP [56]) as a SegCoder and a two-layer MLP [41] which we collectively refer to as our **V**ersatile en**Coder** (**VCoder**). This extra control from the segmentation map results in considerable performance gains on the object identification task.

As shown in Fig. 4a, our VCoder adapted MLLM takes three sets of inputs: perception modalities as control inputs fed into the VCoder, an RGB image fed into an Image enCoder (and MLP), and the question from the user. The RGB image and text are tokenized to the `<img>` and `<query>` tokens, respectively. VCoder is flexible at handling various perception modalities with a special token for each modality. For example, the segmentation map and depth map inputs are tokenized to `<seg>` and `<depth>` tokens, respectively. Similarly, one can incorporate more modalities with modality-specific tokens. Finally, all tokenized embeddings are concatenated and fed into the LLM to obtain the final answer. We only use the `<seg>` input for the object identification task.

We treat our VCoder as an adapter, added to our base MLLM, LLaVA-1.5 [41] to obtain the final MLLM framework for experiments. Note that we only train the MLP components in the VCoder on the COST dataset. We decided to keep all other parameters fixed during training to keep the reasoning ability unaffected while achieving improved object perception performance.

3.3. Evaluating MLLMs for Object Identification

Despite the availability of various metrics [37, 45, 59] to measure object hallucination in vision-language models, no existing metric considers the explicit object counts while calculating their hallucination scores. We argue that object counts returned by an MLLM are a critical component that should not be overlooked while evaluating object identification performance. Therefore, we propose evaluating object identification performance in MLLMs using two metrics: count-score (**CS**) and hallucination-score (**HS**).

$$\begin{aligned} G_{\text{dict}} &= \{\text{OBJ}_1^G : \text{CNT}_1^G; \cdots ; \text{OBJ}_N^G : \text{CNT}_N^G\} \\ P_{\text{dict}} &= \{\text{OBJ}_1^P : \text{CNT}_1^P; \cdots ; \text{OBJ}_M^P : \text{CNT}_M^P\} \end{aligned} \tag{1}$$

As shown in Fig. 5, given a ground-truth sentence (G) and an MLLM predicted response (P), we first extract the object words (nouns) and their corresponding count from both text samples and represent them in a dictionary form

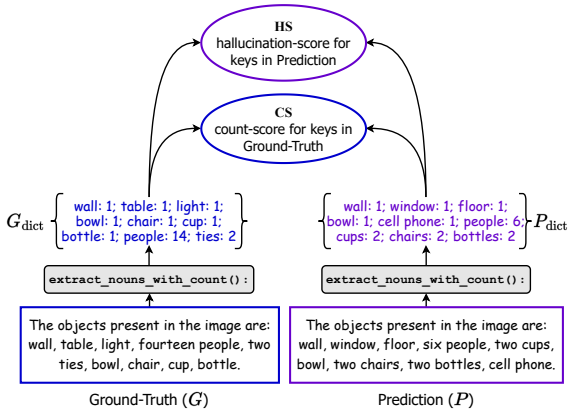


Figure 5. **Evaluation Metrics for Object Identification**. We compare the object counts in the ground truth and prediction to calculate a count score (**CS**) and a hallucination score (**HS**).

with keys as the object noun and the value as the corresponding object's count as shown in Eq. (1) with N and M representing the number of different object nouns in the G and P respectively. Next, we perform one-to-one matching between the counts for keys with G_{dict} and P_{dict} as the reference for Count Score (**CS**) and Hallucination Score (**HS**), respectively, as shown in Eq. (2).

$$\begin{aligned} \textbf{CS} &= \frac{100}{N}\sum_{i=1}^{N}\begin{cases}\frac{\min(\text{CNT}_i^G,\text{CNT}_i^P)}{\max(\text{CNT}_i^G,\text{CNT}_i^P)} & \text{if } I(\text{OBJ}_i^G, P_{\text{dict}}) \\ 0 & \text{otherwise}\end{cases} \\ \textbf{HS} &= \frac{100}{M}\sum_{j=1}^{M}\begin{cases}1-\frac{\min(\text{CNT}_j^P,\text{CNT}_j^G)}{\max(\text{CNT}_j^P,\text{CNT}_j^G)} & \text{if } I(\text{OBJ}_j^P, G_{\text{dict}}) \\ 1 & \text{otherwise}\end{cases} \\ I(\text{OBJ}, D) &= \begin{cases}\text{True} & \text{if OBJ is in } \texttt{keys}(D) \\ \text{False} & \text{otherwise}\end{cases} \end{aligned} \tag{2}$$

Count Score (CS). It represents the percentage of correct object counts predicted by the MLLM with respect to the ground-truth sentence. The higher the **CS**, the better.

Hallucination Score (HS). It represents the percentage of extra object counts predicted by the MLLM that do not exist in the ground-truth sentence. The lower the **HS**, the better.

Note that due to the one-to-one word-matching nature of our evaluation, we manually define a mapping between the categories in COCO [38] and their synonyms [46, 59]. For example, we replace words like *man, woman, child, kid, boy, girl*, *etc.* with the word *person* in the MLLM's response before evaluation.

4. Experiments

We use LLaVA-1.5 [41] as our base MLLM. LLaVA-1.5 uses CLIP-ViT-L-336px [56] as the image encoder (ImCoder) with a two-layer MLP as projection and Vicuna-1.5 [77] as the LLM. Inside our VCoder, we also use a CLIP-ViT-L-336px to encode the control inputs and

Method	Input Tokens	Semantic		Instance		Panoptic	
		CS (↑)	HS (↓)	CS (↑)	HS (↓)	CS (↑)	HS (↓)
Closed Model, Open API							
GPT-4V [51]	⟨*img*⟩ + ⟨*query*⟩	—	—	—	—	38.4	83.0
Existing Open-Source Multimodal LLMs							
MiniGPT-4 LLaMA-2-7b [78]	⟨*img*⟩ + ⟨*query*⟩	6.2	92.2	5.6	97.7	6.2	94.9
InstructBLIP Vicuna-7b [14]	⟨*img*⟩ + ⟨*query*⟩	14.2	85.8	25.3	91.9	17.5	91.2
LLaVA-1.5-7b [41]	⟨*img*⟩ + ⟨*query*⟩	30.6	60.1	50.3	75.9	38.7	67.3
LLaVA-1.5-13b [41]	⟨*img*⟩ + ⟨*query*⟩	25.0	69.3	49.9	75.0	35.8	68.6
CogVLM-17b [68]	⟨*img*⟩ + ⟨*query*⟩	33.4	67.5	43.5	86.2	40.6	75.9
*Baselines trained on the **COST** dataset*							
COST IT LLaVA-1.5-7b	⟨*img*⟩ + ⟨*query*⟩	78.7	22.1	67.5	30.3	71.9	28.2
Soft-Prompted LLaVA-1.5-7b	⟨*prompt*⟩ + ⟨*img*⟩ + ⟨*query*⟩	36.2	56.7	18.4	72.2	26.8	63.0
ImCoder LLaVA-1.5-7b	⟨*img*⟩ + ⟨*img*⟩ + ⟨*query*⟩	78.9	22.7	64.0	29.4	70.8	27.9
***VCoder** augmented LLaVA-1.5*							
VCoder LLaVA-1.5-7b	⟨*seg*⟩ + ⟨*img*⟩ + ⟨*query*⟩	88.6	10.4	71.1	26.9	86.0	12.8
VCoder LLaVA-1.5-13b	⟨*seg*⟩ + ⟨*img*⟩ + ⟨*query*⟩	**89.0**	**10.0**	**73.3**	**25.0**	**87.2**	**11.6**

Table 1. **Comparison to baseline Multimodal LLMs on the COST validation dataset for Object Identification.** We compare our VCoder to existing off-the-shelf baseline MLLMs: MiniGPT-4 [78], InstructBLIP [14], LLaVA-1.5 [41], and CogVLM [68]. We also train three different variants of LLaVA-1.5 on the COST dataset: *COST IT* mixes the COST training data with the instruction tuning data; *Soft-Prompted* uses a set of learnable tokens, and *ImCoder* uses an RGB image as the control input. Our **VCoder** adapted LLaVA-1.5 performs the best on all three object perception tasks. Note: ⟨·⟩ denotes input tokens to LLM with *seg* representing segmentation map, *img* representing RGB image, *prompt* representing learnable prompt, and *query* representing the user question. We also evaluate the performance of GPT-4V [51] on the COST dataset using the publicly accessible paid API released by OpenAI. Our VCoder-adapted LLaVA-1.5 shows the best performance on object identification among all MLLMs.

project the features into the LLM embedding space using modality-specific two-layer MLPs. We resize the visual inputs to 336×336 resolution (corresponds to 576 tokens) for our MLLM. During training, we load the instruction-tuned weights from LLaVA-1.5 and keep those frozen while only tuning the MLP component of our VCoder. We use the publicly available OneFormer [26] model trained on COCO [38] with DiNAT-L [19, 20] backbone to obtain the segmentation map. For getting depth maps, we use the publicly available ViT-L/14 distilled variant of DINOv2 [52] DPT [57] trained on the NYUd [49] dataset. In this section, we discuss our results on the object identification task. Please refer to Sec. 5 for our results on the object order perception task.

4.1. Implementation Details

Training Details. We train our VCoder-adapted LLaVA-1.5 framework for two epochs on the COST training dataset with a batch size 256 and a learning rate of $1e^{-3}$. For other training hyperparameters, we follow the settings used during the instruction-tuning stage in LLaVA-1.5 [41]. Following [26], we uniformly sample each object identification task (semantic, instance, and panoptic) during training. We also use the corresponding segmentation map from OneFormer [26] as input to the VCoder during training and inference. On 8 A100 GPUs, it takes 8 and 14 hours to train our VCoder with the 7b and 13b variants of LLaVA-1.5 as the base MLLM, respectively.

Evaluation Details. We evaluate all MLLMs on the COST validation set. We separately evaluate semantic, instance, and panoptic object identification tasks while randomly sampling questions from the corresponding task's question bucket. Note that for evaluating all off-the-shelf MLLMs, we experiment with various prompts and finally use the prompt: *"[QUESTION]. Return the answer in the paragraph format: 'The objects present in the image are: ...' and then list the objects with their count in word format (if greater than 1) in front of them, like 'two people'."*, where [QUESTION] is the randomly sampled question from the object identification task bucket.

4.2. Main Results

Baselines. We compare the performance of VCoder to open-source Multimodal LLMs, namely, MiniGPT-4 [78], InstructBLIP [14], LLaVA-1.5 [41], and CogVLM [68] on the COST validation set in Tab. 1. Furthermore, we also provide three additional baselines, all trained for two

Method	Depth Score (↓)
LLaVA-1.5-7b [41]	166.1
LLaVA-1.5-13b [41]	227.2
VCoder-DS LLaVA-1.5-7b	**65.9**
VCoder-DS LLaVA-1.5-13b	**63.3**

Table 2. **Performance on Object Order Perception.** Our VCoder LLaVA-1.5 considerably outperforms LLaVA-1.5 [41], owing to the usage of control inputs and training on the COST dataset.

epochs:

COST IT LLaVA-1.5: We mix the COST training data with the instruction tuning data used in LLaVA-1.5 [41] and fine-tune a LLaVA-1.5 model from scratch following the settings from Liu *et al.* [41].

Soft-Prompted LLaVA-1.5: We prepend 576 learnable tokens ($\langle prompt \rangle$) to the LLM input and tune only the $\langle prompt \rangle$ parameters on the COST training dataset.

ImCoder LLaVA-1.5: We use an RGB image as the control input instead of a segmentation map and train VCoder on the COST training dataset.

As shown in Tab. 1, we notice that all existing MLLMs perform poorly on our COST validation set, demonstrating their inability to count and identify objects accurately. Note that existing MLLMs perform relatively better on instance object identification, reaffirming our claim that MLLMs are better at detecting salient objects than background objects. Although the baselines trained on the COST dataset perform relatively better, they still lag in performance compared to the VCoder. Notably, a segmentation map performs considerably better than using an RGB image as the control input, proving the segmentation map's vitality.

Comparison to GPT-4V [51]. We utilize OpenAI's newly released `gpt-4-vision-preview`[1] API to obtain responses from GPT-4V. Our experiments show that GPT-4V's responses are consistent across all object identification tasks, closely aligning with the panoptic identification task. Therefore, we compare our VCoder to GPT-4V only on the panoptic object identification to reduce API requests due to a daily limit of 500 API requests during this project. As shown in Tab. 1, GPT-4V [51] lags behind our VCoder by a considerable margin, reaffirming our claim that existing MLLMs cannot perform accurate object-level perception.

5. Object Order Perception with MLLMs

As shown in Fig. 4, multiple perception modalities can be leveraged to improve object perception in MLLMs with our VCoder. This section presents our experiments with our VCoder using the segmentation and depth maps as the control inputs. We term the resulting MLLM as VCoder-DS LLaVA-1.5. Intuitively, predicting the object order implicitly means identifying the objects in an image. Therefore,

[1] https://platform.openai.com/docs/guides/vision

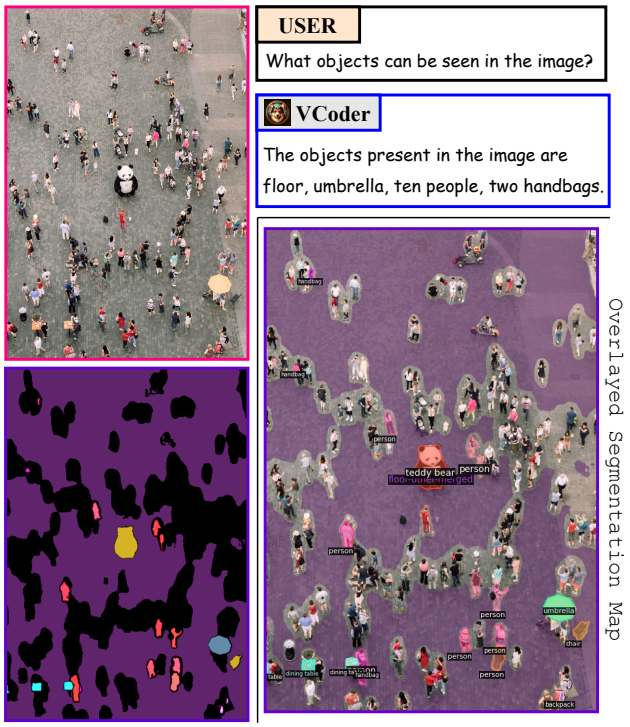


Figure 6. **Failure Case.** VCoder returns the wrong response when the input segmentation mask (control input) is inaccurate.

for the object order perception task (Fig. 4b), we use both `<depth>` and `<seg>` inputs, while only the `<seg>` input as the additional control for object identification.

During training, we use a mixture of datasets, including the object identification and object order perception components from the COST dataset. We also use about 200k image-conversation (along with the corresponding segmentation map obtained using OneFormer [26]) pairs randomly sampled from the instruction tuning data used in LLaVA-1.5 [41]. We train our VCoder for one epoch following the same hyperparameter settings mentioned in Sec. 4.

As shown in Tab. 2, our VCoder-DS LLaVA-1.5 significantly outperforms the base MLLM, LLaVA-1.5 [41] on the COST validation set. For quantitatively evaluating the performance of MLLMs on the depth order perception task, we calculate a depth score (**DS**) using the absolute difference between the position of objects in the ground truth and prediction. We provide more details about computing the depth score in the appendix.

6. Limitations

Despite the improved object perception performance after training our VCoder on the COST dataset, certain limitations remain to be addressed for future work. Firstly, we build our COST dataset using OneFormer [26], which can only perceive objects belonging to a limited number of categories due to being trained on a closed-set vocabulary dataset [38]. For real-world applications, it is impera-

tive to develop an object perception benchmark for MLLMs covering many more classes with varying granularity than those in the COCO [38]. Secondly, the count, hallucination, and depth scores use one-to-one word matching, which requires manually defining a mapping between synonymous words. It will be promising to explore ways to overcome manually defined synonym mappings. Lastly, as shown in Fig. 6, the inaccuracy in the segmentation map may result in the VCoder's failure. Exploring ways to reduce the over-dependency on control inputs to handle inaccurate context from the perception modalities would be interesting.

7. Conclusion

This work analyzes the object-level perception skills of Multimodal Large Language Models (VLMMs). Although MLLMs are good visual reasoners, they need to improve at the simple yet fundamental task of object perception. To improve object perception ability in MLLMs, we propose the COST dataset for training and evaluating MLLMs at the object perception task. We benchmark different off-the-shelf MLLMs and GPT-4V on our COST dataset and observe their lousy performance. Consequently, we propose using perception modalities as control inputs and a Versatile vision enCoders (**VCoder**) as an adapter for projecting the control inputs to the LLM embedding space. Our VCoder can easily be extended to leverage various modalities as the control inputs depending on the task. To quantify the object-level perception ability in MLLMs, we introduce a Count-Score (**CS**), a Hallucination-Score (**HS**), and a Depth-Score (**DS**). We adapted LLaVA-1.5 with VCoder, only trained the VCoder on our COST dataset, and demonstrated its improved performance at the object perception task while retaining the reasoning performance. We hope our work can inspire the research community to focus on developing object perception datasets for MLLMs and develop vision systems that are equally good at perception and reasoning in the future.

Acknowledgements. We would like to extend our gratitude to Eric Zhang and Kai Wang (JJ's labmates) for an insightful discussion before the start of the project and valuable feedback on the design of Figure 2. We also thank the ML Center at Georgia Tech for generously supporting this work.

References

[1] Aishwarya Agrawal, Jiasen Lu, Stanislaw Antol, Margaret Mitchell, C. Lawrence Zitnick, Dhruv Batra, and Devi Parikh. Vqa: Visual question answering, 2015. 2

[2] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. Flamingo: a visual language model for few-shot learning. In *NeurIPS*, 2022. 1, 3

[3] Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Merouane Debbah, Etienne Goffinet, Daniel Heslow, Julien Launay, Quentin Malartic, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. Falcon-40B: an open large language model with state-of-the-art performance. *arXiv*, 2023. 1

[4] Anas Awadalla, Irena Gao, Josh Gardner, Jack Hessel, Yusuf Hanafy, Wanrong Zhu, Kalyani Marathe, Yonatan Bitton, Samir Gadre, Shiori Sagawa, Jenia Jitsev, Simon Kornblith, Pang Wei Koh, Gabriel Ilharco, Mitchell Wortsman, and Ludwig Schmidt. Openflamingo: An open-source framework for training large autoregressive vision-language models. *arXiv*, 2023. 1, 3

[5] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. Qwen technical report. *arXiv*, 2023. 3

[6] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020. 3

[7] Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. Conceptual 12M: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *CVPR*, 2021. 2

[8] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. In *ICLR*, 2015. 3

[9] Bowen Cheng, Maxwell D Collins, Yukun Zhu, Ting Liu, Thomas S Huang, Hartwig Adam, and Liang-Chieh Chen. Panoptic-deeplab: A simple, strong, and fast baseline for bottom-up panoptic segmentation. In *CVPR*, 2020. 3

[10] Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *CVPR*, 2022. 3, 5, 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, 3

[12] OpenCompass Contributors. Opencompass: A universal evaluation platform for foundation models. https://

Summary

- One of the key ideas for using multimodal models is to provide context beyond a single modality so that the model performs better on tasks where contextual information is critical.
- Vision-language models are one of the kinds of MLLM among the hottest trends since last year.
- Though Multimodal Large Language Models (MLLM) have recently achieved impressive performance on vision-language tasks ranging from visual question-answering and image captioning to visual reasoning and image generation; when prompted to identify or count (perceive) the entities in a given image, existing MLLM systems fail. Counting objects in a given image is one such task. Not only do the existing models fail to count objects accurately, but they also hallucinate about certain non-existing objects in a given image. To solve perception and reasoning, we need better VLMs.
- The key idea presented in this paper is to improve the object-level perception of a VLM because the authors believe that one can only describe and reason about a visual scene with the correct understanding of the entities in the image.
- Sounds easy, but there are major challenges to solving the above problem like:
    - The scarcity of a vision-language dataset focused on the object perception task.
    - Existing open-sourced Multimodal LLMs usually use the ViT from CLIP with an RGB image as input as the visual component that focuses only on salient objects.
    - The absence of evaluation metrics to quantitatively measure Multimodal LLMs' object perception, especially counting skills.

Related Work

- Perception can be divided into sub-tasks, including dense prediction tasks like image segmentation and depth estimation, and sparse prediction tasks like object detection and pose estimation.
- The above tasks can be solved easily with any CNN/Transformers-based network. The authors in this paper focus on object-level perception focusing on predicting names, counts, and order of objects in an image using MLLMs.
- Developing a VLM generally involves projecting the features from a vision encoder to the embedding space of a language model, and performing instruction-tuning on a vision language dialog dataset.
- Different models do this in different ways, and the authors picked LLaVa 1.5 as the baseline because of its superior performance.

Dataset Curation

- The authors hypothesize that the main reason existing VLMs fail on simple tasks like object counting is the absence of conversations covering object identification for not only the salient objects but also the objects in the background from the instruction tuning data for MLLMs.
- To overcome this issue, the authors prepare the COCO Segmentation Text (COST) dataset with COCO images and create sentences using the output from an image segmentation model to obtain an image-text dataset to train and evaluate MLLMs for object perception MLLMs. They also introduce a segmentation map as a control image input to the MLLM for better performance and quantify object perception performance with a count score and a hallucination score.

- The segmentation maps are obtained using the OneFormer segmentation model. They then extract the object (class) names and counts from the segmentation outputs and convert them into a sentence form for the ground-truth answer: “The objects present in the image are: [CNT1] [OBJ1], [CNT2] [OBJ2], . . ., [CNTN] [OBJN].”, with [OBJi] representing the object name and [CNTi] representing the count (if greater than one) for the ith object in the image.
- Then the authors used GPT-4 to collect a pool of questions related to three different object identification tasks: semantic, instance, and panoptic segmentation
- The original image, the segmentation maps, and the pooled questions are combined to form the COCO Segmentation Text (COST) dataset for training and evaluating MLLMs on the object identification task.
- The authors also extend COST to include other perception modalities like depth map modality. They used DINOv2 to obtain the depth maps, and corresponding texts are generated in the form: "The depth order for objects present in the image is: [OBJ1], [OBJ2], . . ., [OBJJ].”, with [OBJj] representing the j th object name."
- Similar to the segmentation maps, the authors pooled questions by prompting GPT-4 for the object order perception task.

VCoder

- Existing VLMs use ViT from CLIP as the image encoder. Given that it is trained against captions, the authors believe it focuses more on the salient objects and ignores the background. I do not agree with this take as generating captions requires an understanding of different aspects of an image.
- To overcome this limitation, they introduce a segmentation map as a control input. Specifically, they use the segmentation map from OneFormer and project it to the LLM’s embedding space using a pre-trained ViT from CLIP as a SegCoder, and a two-layer MLP. This setup is what they refer to as a Versatile Encoder, or VCoder for short. The control input doesn't need to be limited to only a segmentation map. You can add as many perception modalities as you like and use a similar projection setup.
- VCoder adapted MLLM takes three sets of inputs:
    - Perception modalities as control inputs fed into the VCoder
    - An RGB image fed into an Image enCoder(ViT from CLIP) (and MLP)
    - The question from the user. The RGB image and text are tokenized to the <img> and <query> tokens respectively.
- You can add a specialized token for each perception modality, like <seg> to denote the segmentation map.
- This Vcoder is then added to the base LLaVa 1.5 model. Only MLP layers of the Vcoder are trainable.

Evaluation: Count Score and Hallucination Score

- No existing metric considers the explicit object counts while calculating their hallucination scores.
- Count Score represents the percentage of correct object counts predicted by the MLLM concerning the ground-truth sentence. The higher the count score, the better the performance.
- Hallucination Score represents the percentage of extra object counts predicted by the MLLM that do not exist in the ground-truth sentence. The lower the hallucination score, the better the model.

github.com/open-compass/opencompass, 2023. 3

[13] Timothy F Cootes, Gareth J Edwards, and Christopher J Taylor. Active appearance models. *IEEE Transactions on pattern analysis and machine intelligence*, 23(6):681–685, 2001. 3

[14] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning, 2023. 1, 2, 3, 7

[15] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 3, 5, 6

[16] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Advances in neural information processing systems*, pages 2366–2374, 2014. 3

[17] Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Zhenyu Qiu, Wei Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, and Rongrong Ji. Mme: A comprehensive evaluation benchmark for multimodal large language models. *arXiv*, 2023. 3

[18] Rohit Girdhar, Mannat Singh, Nikhila Ravi, Laurens van der Maaten, Armand Joulin, and Ishan Misra. Omnivore: A Single Model for Many Visual Modalities. In *CVPR*, 2022. 3

[19] Ali Hassani and Humphrey Shi. Dilated neighborhood attention transformer. *arXiv:2209.15001*, 2022. 7

[20] Ali Hassani, Steven Walton, Jiachen Li, Shen Li, and Humphrey Shi. Neighborhood attention transformer. In *CVPR*, 2023. 7

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1

[22] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017. 3, 1

[23] Shaohan Huang, Li Dong, Wenhui Wang, Yaru Hao, Saksham Singhal, Shuming Ma, Tengchao Lv, Lei Cui, Owais Khan Mohammed, Qiang Liu, Kriti Aggarwal, Zewen Chi, Johan Bjorck, Vishrav Chaudhary, Subhojit Som, Xia Song, and Furu Wei. Language is not all you need: Aligning perception with language models. *ArXiv*, abs/2302.14045, 2023. 1, 3

[24] Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. *CVPR*, 2019. 3

[25] Jitesh Jain, Anukriti Singh, Nikita Orlov, Zilong Huang, Jiachen Li, Steven Walton, and Humphrey Shi. Semask: Semantically masking transformer backbones for effective semantic segmentation. *arXiv*, 2021. 3

[26] Jitesh Jain, Jiachen Li, MangTik Chiu, Ali Hassani, Nikita Orlov, and Humphrey Shi. OneFormer: One Transformer to Rule Universal Image Segmentation. In *CVPR*, 2023. 2, 3, 4, 5, 6, 7, 8, 1

[27] Yang Jin, Kun Xu, Kun Xu, Liwei Chen, Chao Liao, Jianchao Tan, Yadong Mu, et al. Unified language-vision pre-training in llm with dynamic discrete visual tokenization. *arXiv*, 2023. 1

[28] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. In *CVPR*, 2019. 1

[29] Jing Yu Koh, Daniel Fried, and Ruslan Salakhutdinov. Generating images with multimodal language models. *NeurIPS*, 2023. 1

[30] Deanna Kuhn. *The Skills of Argument*. Cambridge University Press, 1991. 1

[31] Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab. Deeper depth prediction with fully convolutional residual networks. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 239–248. IEEE, 2016. 3

[32] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*. 3

[33] Bo Li, Yuanhan Zhang, Liangyu Chen, Jinghao Wang, Fanyi Pu, Jingkang Yang, Chunyuan Li, and Ziwei Liu. Mimic-it: Multi-modal in-context instruction tuning. 2023. 3

[34] Bo Li, Yuanhan Zhang, Liangyu Chen, Jinghao Wang, Jingkang Yang, and Ziwei Liu. Otter: A multi-modal model with in-context instruction tuning. *arXiv preprint arXiv:2305.03726*, 2023. 3

[35] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *ICML*, 2022. 2

[36] 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. 2, 3

[37] 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. 3, 6

[38] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context. In *ECCV*, 2014. 2, 4, 5, 6, 7, 8, 9

[39] Fuxiao Liu, Tianrui Guan, Zongxia Li, Lichang Chen, Yaser Yacoob, Dinesh Manocha, and Tianyi Zhou. Hallusionbench: You see what you think? or you think what you see? an image-context reasoning benchmark challenging for gpt-4v (ision), llava-1.5, and other multi-modality models. *arXiv preprint arXiv:2310.14566*, 2023. 3

[40] Fuxiao Liu, Kevin Lin, Linjie Li, Jianfeng Wang, Yaser Yacoob, and Lijuan Wang. Aligning large multi-modal model with robust instruction tuning. *arXiv preprint arXiv:2306.14565*, 2023. 3

[41] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning, 2023. 1, 2, 3, 5, 6, 7, 8

[42] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023. 1, 3

[43] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021. 1

[44] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 3

[45] Holy Lovenia, Wenliang Dai, Samuel Cahyawijaya, Ziwei Ji, and Pascale Fung. Negative object presence evaluation (nope) to measure object hallucination in vision-language models. *arXiv*, 2023. 6

[46] Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. Neural baby talk. In *CVPR*, 2018. 6

[47] H. Moravec. Mind children: The future of robot and human intelligence. Harvard University Press, 1988. 4

[48] Chong Mou, Xintao Wang, Liangbin Xie, Yanze Wu, Jian Zhang, Zhongang Qi, Ying Shan, and Xiaohu Qie. T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. *arXiv*, 2023. 3, 6

[49] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012. 7, 1

[50] OpenAI. Chatgpt. https://chat.openai.com/, 2022. 1

[51] OpenAI. Gpt-4 technical report, 2023. 1, 2, 3, 4, 5, 7, 8

[52] Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2023. 2, 4, 5, 7, 1

[53] Vicente Ordonez, Girish Kulkarni, and Tamara Berg. Im2text: Describing images using 1 million captioned photographs. In *NeurIPS*, 2011. 2

[54] 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*, abs/2306, 2023. 1

[55] Alec Radford, Jong Wook Kim, Chris Hallacy, A. Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 2, 3

[56] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. *arXiv*, 2021. 3, 5, 6

[57] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *ICCV*, 2021. 2, 3, 4, 5, 7, 1

[58] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. *arXiv*, 2015. 3

[59] Anna Rohrbach, Lisa Anne Hendricks, Kaylee Burns, Trevor Darrell, and Kate Saenko. Object hallucination in image captioning. In *EMNLP*, 2018. 3, 6

[60] 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. In *NeurIPS Workshops 2021*, 2021. 2

[61] Quan Sun, Qiying Yu, Yufeng Cui, Fan Zhang, Xiaosong Zhang, Yueze Wang, Hongcheng Gao, Jingjing Liu, Tiejun Huang, and Xinlong Wang. Generative pretraining in multimodality. *arXiv*, 2023. 1

[62] Alexander Toshev and Christian Szegedy. Deeppose: Human pose estimation via deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1653–1660, 2014. 3

[63] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models. *arXiv*, 2023. 3

[64] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models. *arXiv*, 2023. 1, 3

[65] Maria Tsimpoukelli, Jacob Menick, Serkan Cabi, S. M. Ali Eslami, Oriol Vinyals, and Felix Hill. Multimodal few-shot learning with frozen language models. In *NeurIPS*, 2021. 3

[66] Z. Tu, Xiangrong Chen, Alan Yuille, and Song Zhu. Image parsing: Unifying segmentation, detection, and recognition. In *IJCV*, 2005. 3

[67] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *CVPR*, 2001. 3

[68] Weihan Wang, Qingsong Lv, Wenmeng Yu, Wenyi Hong, Ji Qi, Yan Wang, Junhui Ji, Zhuoyi Yang, Lei Zhao, Xixuan Song, Jiazheng Xu, Bin Xu, Juanzi Li, Yuxiao Dong, Ming Ding, and Jie Tang. Cogvlm: Visual expert for pretrained language models. *arXiv*, 2023. 7

[69] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M. Alvarez, and Ping Luo. Segformer: Simple and ef-

ficient design for semantic segmentation with transformers. In *NeurIPS*, 2021. 3

[70] Xingqian Xu, Jiayi Guo, Zhangyang Wang, Gao Huang, Irfan Essa, and Humphrey Shi. Prompt-free diffusion: Taking" text" out of text-to-image diffusion models. *arXiv preprint arXiv:2305.16223*, 2023. 3

[71] Xingqian Xu, Zhangyang Wang, Eric Zhang, Kai Wang, and Humphrey Shi. Versatile diffusion: Text, images and variations all in one diffusion model. In *ICCV*, 2023. 3

[72] Qinghao Ye, Haiyang Xu, Guohai Xu, Jiabo Ye, Ming Yan, Yiyang Zhou, Junyang Wang, Anwen Hu, Pengcheng Shi, Yaya Shi, Chaoya Jiang, Chenliang Li, Yuanhong Xu, Hehong Chen, Junfeng Tian, Qian Qi, Ji Zhang, and Fei Huang. mplug-owl: Modularization empowers large language models with multimodality, 2023. 1, 3

[73] Yan Zeng, Hanbo Zhang, Jiani Zheng, Jiangnan Xia, Guoqiang Wei, Yang Wei, Yuchen Zhang, and Tao Kong. What matters in training a gpt4-style language model with multimodal inputs? *arXiv preprint arXiv:2307.02469*, 2023. 3

[74] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *ICCV*, 2023. 3, 6

[75] Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, Longyue Wang, Anh Tuan Luu, Wei Bi, Freda Shi, and Shuming Shi. Siren's song in the ai ocean: A survey on hallucination in large language models. *arXiv*, 2023. 3

[76] Kaizhi Zheng, Xuehai He, and Xin Eric Wang. Minigpt-5: Interleaved vision-and-language generation via generative vokens. *arXiv*, 2023. 1

[77] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric. P Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena, 2023. 6

[78] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv*, 2023. 1, 3, 7

[79] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv*, 2020. 3

Appendix

In this appendix, we first present our analysis of the effect of the quality of the segmentation map (control input) on the VCoder's performance in Appendix A. Next, we provide details about obtaining ground-truth texts for the object order perception task along with the process to compute the depth score in Appendix B. Lastly, we share analysis on the per-image object counts about the COST dataset in Appendix C.

A. Control Through Segmentation Map

We study the effect of segmentation map quality on object identification performance. Specifically, instead of using DiNAT-L OneFormer [26] to obtain the segmentation map, we use the relatively worse segmentation models: ResNet-50 [21] based Mask R-CNN [22], Panoptic-FPN [28], and Swin-L [43] based Mask2Former [10] for the instance and panoptic object identification task, respectively. As shown in Tab. I, we notice a considerable drop in performance with maps from Mask R-CNN and Panoptic FPN. However, the drop in performance is much lower with maps from a relatively newer and better Mask2Former model, demonstrating the importance of the segmentation map's quality.

Seg Model	Year	CS (↑)	HS (↓)
Instance Object Identification			
OneFormer [26]	CVPR 2023	**71.1**	**26.9**
Mask R-CNN [22]	ICCV 2017	61.9 (-9.2)	39.8 (+12.9)
Panoptic Object Identification			
OneFormer [26]	CVPR 2023	**86.0**	**12.8**
Mask2Former [10]	CVPR 2022	76.5 (-9.5)	26.1 (+13.3)
Panoptic FPN [28]	CVPR 2019	64.2 (-21.8)	33.3 (+20.5)

Table I. **Ablation on Quality of Segmentation Map**. Using segmentation maps from older models like Mask R-CNN [22] and Panoptic-FPN [28] as the control input results in a performance drop due to the relatively low quality of the maps.

B. Object Order Perception

In this section, we present the process of obtaining the ground truth ordering of objects in an image using segmentation and depth maps. Then, we share details about the logic used to compute the depth score (**DS**).

B.1. Obtaining Ground Truth

To obtain the ground truth order for objects in an image, we utilize the fact that each pixel in a depth map (from DINOv2 [52] DPT [57]) represents the distance [49] of that pixel from the camera. Therefore, as shown in Fig. I, we use the binary object masks (from OneFormer's [26] panoptic prediction) to first obtain the corresponding regions in the depth map. Next, for each object region, we calculate the maximum pixel value representing the distance of the object's farthest point from the camera. Finally, we sort the values obtained in the previous in an ascending order to obtain the final order, starting with the closest object and ending with the farthest object. As mentioned in Sec. 5, we append a number to the object name to represent the relative order of objects belonging to the same category.

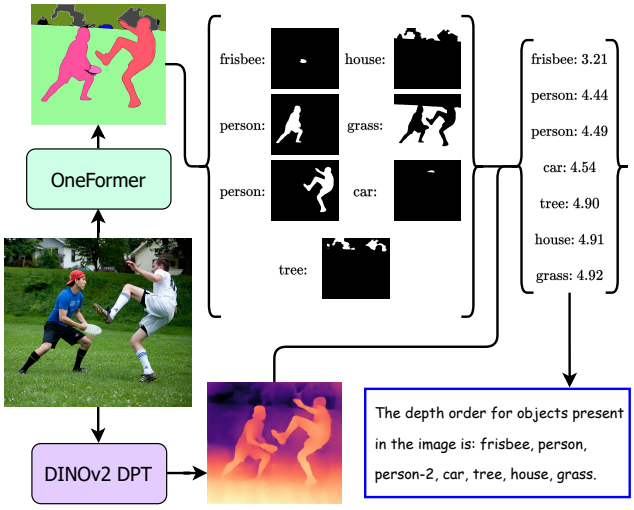


Figure I. **Data Engine to obtain Object Order GT**. We calculate the maximum pixel value inside each object's region using the depth and segmentation maps. We sort the obtained values in an ascending order to obtain the final object order.

B.2. Depth Score

In Fig. II, we share the `python` code to compute the depth score given the ground truth and prediction for object orders in an image. Particularly, we first obtain the position of objects belonging to all categories and then compute the absolute difference using the position values for objects belonging to the same category in the ground truth and prediction. Note that to handle different numbers of objects in the prediction and ground truth, we use the position value as 100 for unmatched objects. We average the obtained score over all images to obtain the final depth score.

C. Object Counts in COST Dataset

We show the plots for the per-image total object count distribution in the `train` and `val` splits of our COST dataset in Fig. III. We observe that there exists a long tail beyond the object count of 25. Based on this observation, we express the need for a more scaled effort at collecting object-level perception datasets for training MLLMs to make them excel (without extra pre-processing) at counting in cluttered scenes that may contain many more objects.

```python
def calculate_per_image_depth_score(gt, pred):
    position_gt, order_num = _get_order(gt)
    position_pred, _ = _get_order(pred)
    depth_distance = []

    for object in position_gt.keys():
        if position_pred is not None and object in position_pred.keys():
            order_pred = position_pred[object]
            order_gt = position_gt[object]
            # pad the object specific position list to make with 100 to make them equal for prediction and ground-truth
            if len(order_gt) < len(order_pred):
                order_gt.extend([100] * (len(order_pred) - len(order_gt)))
            elif len(order_pred) < len(order_gt):
                order_pred.extend([100] * (len(order_gt) - len(order_pred)))
            for i, j in zip(order_gt, order_pred):
                depth_distance.append(abs(i - j))
        else:
            depth_distance.append(100)
    # normalize the score based on the total number of objects in the image
    return sum(depth_distance) / order_num

# helper function to calculate the order position of the objects in the image
def _get_order(text):
    order_num = 1  # order number of the object
    positions = {}
    # obtain object nouns
    nouns = _obtain_nouns(text)
    for noun in nouns:
        # obtain only object noun (person) from words like person-2
        object = noun.split("-")[0].strip()
        if object not in positions.keys():
            positions[object] = [order_num]
        else:
            positions[object].append(order_num)
        order_num += 1
    return positions, order_num - 1
```

Figure II. Computing Depth Score for a given Image.

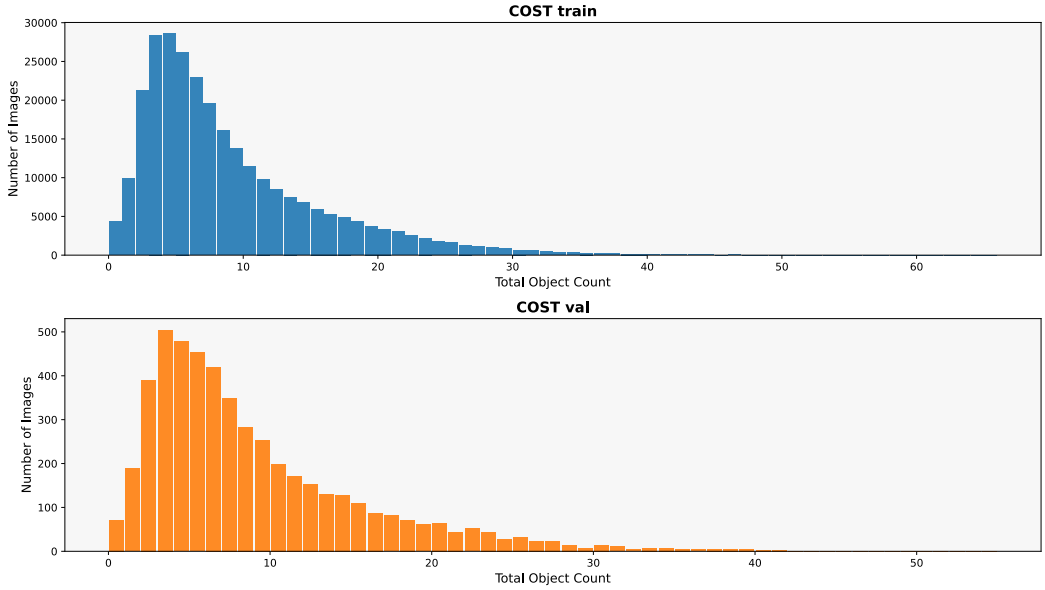


Figure III. **Total Object Counts per image in the COST `train` and `val` splits.** We observe that our COST dataset does not include images with more than 60 objects and has a long tail beyond the object count of 25.