OLIVE: Object Level In-Context Visual Embeddings

Timothy Ossowski[1], **Junjie Hu**[1,2]
[1]Department of Computer Science, [2]Department of Biostatistics and Medical Informatics
University of Wisconsin, Madison, WI, USA
ossowski@wisc.edu, junjie.hu@wisc.edu

Abstract

Recent generalist vision-language models (VLMs) have demonstrated impressive reasoning capabilities across diverse multimodal tasks. However, these models still struggle with fine-grained object level understanding and grounding. In terms of modeling, existing VLMs implicitly align text tokens with image patch tokens, which is ineffective for embedding alignment at the same granularity and inevitably introduces noisy spurious background features. Additionally, these models struggle when generalizing to unseen visual concepts and may not be reliable for domain-specific tasks without further fine-tuning. To address these limitations, we propose a novel method to prompt large language models with in-context visual object vectors, thereby enabling controllable object level reasoning. This eliminates the necessity of fusing a lengthy array of image patch features and significantly speeds up training. Furthermore, we propose region-level retrieval using our object representations, facilitating rapid adaptation to new objects without additional training. Our experiments reveal that our method achieves competitive referring object classification and captioning performance, while also offering zero-shot generalization and robustness to visually challenging contexts.[1]

1 Introduction

Despite the popularity, many existing VLMs such as LLaVA (Liu et al., 2023), MiniGPT4 (Zhu et al., 2023a), and mPLUG-OWL (Ye et al., 2023) handle the entire image for visual understanding, leading to two major shortcomings. First, these VLMs use a visual transformer to split an image into a grid of image patches and embed them into a lengthy array of image patch embeddings that have object level features scattered around different positions of the array. This leads to the different granularity between the image patch tokens and text tokens, further creating difficulty in aligning and grounding visual objects to text concepts. Second, feeding all image patch embeddings to the large language model (LLM) decoder is problematic due to the resulting long context and inefficiency of including in-context examples from multiple images.

To improve fine-grained visual alignment, recent region-based VLMs are pre-trained to integrate object level information into the LLM decoder. GPT4ROI (Zhang et al., 2023b) pre-trains LLMs to understand ROIAlign features (He et al., 2017) extracted from bounding boxes. Other similar methods such as Shikra (Chen et al., 2023) or Kosmos-2 (Peng et al., 2024) ground and refer to objects using text in multimodal referential dialogues. FERRET (You et al., 2023) and ViP-LLaVA (Cai et al., 2023) further support free-form shapes as referring input by summarizing visual features sampled within the region of interest. Although these methods provide improvement to object level reasoning, they still fail at recognizing unseen/rare objects and are sensitive to spurious background features, as shown in §5. Even powerful closed-source multimodal models such as GPT4V are unreliable to deploy in high-stakes domain-specific situations such as the medical domain (Senkaiahliyan et al., 2023).

A straightforward way to handle generalization to unseen visual content is to integrate a retrieval component. Methods such as REVEAL (Hu et al., 2023) and MuRAG (Chen et al., 2022) provide retrieved multimodal facts as supplementary context to help VLMs generalize to new concepts without further training. However, these models do not consider object level retrieval and in-context prediction. Models such as Flamingo (Alayrac et al., 2022) and Qwen-VL (Bai et al., 2023) allow for in-context examples from multiple images, yet do not support object level retrieval and reasoning.

To address the above issues, we propose to encode object level in-context visual embeddings

[1] Our code and models are available at https://github.com/tossowski/OLIVE

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 5170–5185
August 11-16, 2024 ©2024 Association for Computational Linguistics

(OLIVE) to enhance LLMs with region-level reasoning capabilities. Critically, we omit lengthy image patch features and encode visual object embeddings by a lightweight encoder of 20 million parameters, allowing for faster training and direct connection to existing LLMs. This preserves the full functionality of the original LLMs, while also introducing novel multimodal reasoning abilities. Furthermore, our object level retrieval module allows for more precise queries and retrieved information to help the model adapt to domain-specific tasks with limited training data. Our contributions are summarized below and in Table 1:

- We propose a lightweight object encoder that can be connected to existing LLMs to enable controllable object level multimodal reasoning with free-form input annotations.
- Our model omits image patch features and summarizes object features into a single vector, significantly reducing context length for more efficient training and inference, and allowing for in-context examples from multiple images.
- We conduct extensive experiments with region-retrieval of object level features and showcase rapid adaptation to unseen visual concepts.

2 Preliminaries

Generative VLM Architecture Recent generative VLMs (e.g., LLaVA, BLIP-2) adopt a similar architecture that connects a pre-trained visual encoder ϕ_v and a pre-trained language model decoder ϕ_t through a lightweight fusion neural network, denoted as ϕ_c. Specifically, the fusion module first uses a projection function to map a visual feature $\mathbf{v} \in \mathcal{V}$ to the text embedding space $\mathcal{X}$ of the language model decoder, and then fuse the visual and text embeddings as input to language model decoder. Formally, given an image v and a text prompt x, the decoder takes in the combined feature $\mathbf{x}$ to autoregressively predict the output y.

$$\mathbf{x}_t = \texttt{TxtEmbed}(x; \phi_t) \in \mathcal{X} \tag{1}$$

$$\mathbf{v} = \texttt{ImgEncoder}(v; \phi_v) \in \mathcal{V} \tag{2}$$

$$\mathbf{x}_v = \texttt{Project}(\mathbf{v}; \phi_c) \in \mathcal{X} \tag{3}$$

$$\mathbf{x} = \texttt{Fuse}(\mathbf{x}_v, \mathbf{x}_t; \phi_c) \tag{4}$$

$$p_{\text{vlm}}(y|v, x) = \prod_{j=0}^{|y|} p_{\phi_t}(y_j|\mathbf{x}, y_{<j}) \tag{5}$$

Different from prior fusion modules (e.g., linear projection in LLaVA, gated cross-attention in Flamingo, and Q-former in BLIP-2) that project the whole image features, we propose an object level encoder (§3.1) that captures fine-grained region features and speeds up training and inference.

Visual Instruction Tuning We adopt a similar visual instruction-tuning approach as Liu et al. (2023) by fine-tuning parts of the VLM parameters (e.g., ϕ_c and/or ϕ_t) on instruction-following data. The training objective is based on maximum likelihood estimation for next-token predictions given the input image and the text prompt. Different from prior work using pure text prompts, our object encoder and retrieval module (§3.1, §3.2) enables the usage of code-switched prompt sequence mixing text tokens and image object tokens, and the rapid adaptation to unseen domains via in-context prediction.

3 Method

This section as well as Figure 1 highlights the main components of our method. We first design an object encoder (§3.1) to learn visual object embeddings in a shared vision-text space, then apply a similarity search over object embeddings to retrieve relevant visual objects (§3.2), and finally construct a code-switch multimodal prompt to integrate the retrieved object information for generation (§3.3).

3.1 Object Encoder

Following popular region-grounded models such as FERRET (You et al., 2023), we allow for free-form annotation of objects using the object segmentation mask $\mathbf{o}_{\text{mask}}$ as input. Specifically, we first encode an image v with a vision transformer (Dosovitskiy et al., 2020) to obtain patch-level features $\mathbf{v}$:

$$\mathbf{v} = \texttt{ImgEncoder}(v; \phi_v) \in \mathbb{R}^{(n^2+1)\times d}, \tag{6}$$

where n is the grid size and d is the dimension of hidden states. To further obtain an object level feature $\mathbf{v}_{\text{obj}}$ from the image, we first extract a subset of the image features $\mathbf{v}_{\text{masked}}$ corresponding to the binary object segmentation mask $\mathbf{o}_{\text{mask}}$:

$$\mathbf{v}_{\text{masked}} = \mathbf{v}[\texttt{Flatten}(\mathbf{o}_{\text{mask}})] \in \mathbb{R}^{l\times d} \tag{7}$$

where $\mathbf{o}_{\text{mask}}$ is a $n \times n$ binary matrix, indicating the corresponding image patches occupied by an object in the image, and l denotes the number of the occupied patches. These segmentation masks can be created by automatic segmentation tools such as SAM (Kirillov et al., 2023) or provided by human selection on the image. The segmentation

Model	Free-form Visual Prompts	Free-form Text prompts	Visual Generalization	Generative Approach	Multi-Image
Ferret	✓	✓	✗	✓	✗
Flamingo	✗	✓	✓	✓	✓
GPT4ROI	✗	✓	✗	✓	✗
GLAMM	✗	✓	✗	✓	✗
RegionCLIP	✗	✓	✗	✗	✗
Llama-Adapter v2	✗	✗	✗	✓	✗
ViP-LLAVA	✓	✗	✗	✓	✗
OLIVE	✓	✓	✓	✓	✓

Table 1: Comparison of OLIVE to recent VLMs. To the best of our knowledge, we are the first method to offer visual generalization with in-context prompting, while also allowing for free form annotation. A more comprehensive summary of related studies is in §6.

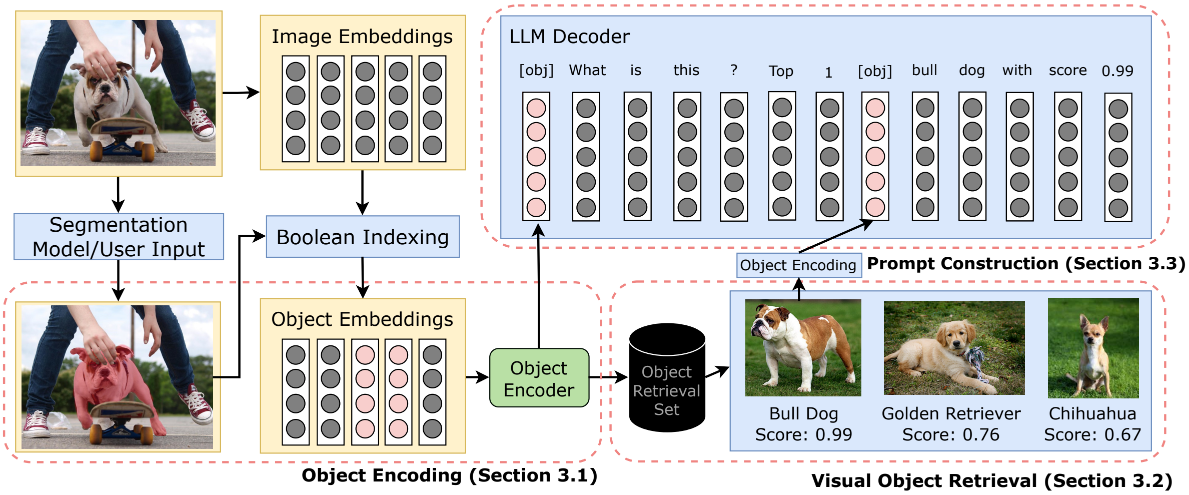


Figure 1: An overview of our method which consists of three main components, described in detail in Section 3. The object encoder (green) is the only module with required trainable parameters. Note: the prompt in the LLM decoder is modified slightly for visual clarity. The exact prompt can be found in Appendix A.

mask is first flattened and used to select object patches $\mathbf{v}_{\text{masked}}$ from $\mathbf{v}$. Finally, we obtain the object embedding by compressing $\mathbf{v}_{\text{masked}}$ into a single vector $\mathbf{v}_{\text{obj}}$.

$$\mathbf{v}_{\text{obj}} = \texttt{ObjectEncoder}(\mathbf{v}_{\text{masked}}; \phi_c) \in \mathbb{R}^d, \quad (8)$$

where the object encoder uses a lightweight 2-layer transformer that acts similar to a visual resampler (Zou et al., 2023a; Li et al., 2023), followed by a learnable linear layer to further project the visual representation to the text space (Liu et al., 2023).

3.2 Visual Object Retrieval

In many cases, the object of interest does not resemble anything seen during training. With our visual object embeddings, we can easily perform object level retrieval to match an open class of visual objects and integrate the retrieved information into the language decoder for predicting unseen or rare objects from specific domains (e.g., biomedicine). To this end, we assume access to a retrieval set $\mathcal{R} = \{(\mathbf{o}_i, d_i, v_i)\}_{i=1}^{m}$, where each triple consists of an object's segmentation mask $\mathbf{o}_i$, the object's text description d_i and the image v_i containing this object. To retrieve relevant objects from $\mathcal{R}$, we use a similar object encoding as §3.1 except that we use the mean pooling of $\mathbf{v}_{\text{masked}}$ as the object encoder in Eq. (9), since this simple strategy does not require any learnable parameters for projection to the text embedding space and visual object embeddings can be pre-computed before any fine-tuning. However, we use a learnable object encoder in Eq. (8) to connect object embeddings to the LM decoder during instruction-tuning for text generation (§3.3).

$$\mathbf{v}_{\text{obj}} = \texttt{MeanPool}(\mathbf{v}_{\text{masked}}) \in \mathbb{R}^d, \quad (9)$$

During retrieval, we compute a query vector $\mathbf{v}_{\text{query}}$ for a given object, and compute the cosine similarity scores between $\mathbf{v}_{\text{query}}$ and all the visual object embeddings from $\mathcal{R}$ to obtain the top k closest triples, denoted as $\mathcal{K} = \{(\mathbf{o}_i, d_i, v_i)\}_{i=1}^{k}$.

3.3 In-context Prompt Construction

As the visual object embeddings are projected into the text embedding space of the LM decoder, this allows us to construct a code-switched prompt that mixes visual objects with text tokens for the LM decoder (e.g., Llama 2 (Touvron et al., 2023)). In addition, as our object encoder compresses a visual object into a single vector $\mathbf{v}_{\text{obj}}$, this significantly shortens the length of the visual tokens that the LM decoder needs to fuse with text tokens. Therefore, we can easily integrate multiple retrieved object embeddings into the prompt to augment the LM decoder for in-context text generation. Specifically, we define a special vocabulary token `[obj]` which can be inserted flexibly in the user prompt x. For example, the user can ask "`[obj]` Describe this part of the image" to perform region-level description. The embedding of this token is directly replaced with its corresponding visual object embedding. Formally, given a text prompt x that contains indexed `[obj]` tokens referring to an object $\mathbf{v}_{\text{obj}}$ of interest in an image v and its relevant objects in $\mathcal{K}$, we define a prompting function that replaces the text embedding of `[obj]` with its corresponding visual object embedding, and integrates the top k most similar objects $\mathcal{K}$ as in-context examples. For example, a prompt with retrieved in-context examples can be "The top [k] related objects are: [obj_1] is a [label],...[obj_k] is a [label]. [obj_query] What is this?". We provide more details about in-context prompt templates and construction in Appendix A.

$$\mathbf{x} = \texttt{Prompt}(x, \mathbf{v}_{\text{obj}}, \mathcal{K}) \tag{10}$$

Finally, we feed the multimodal prompt $\mathbf{x}$ into the LM decoder for text generation following Eq. (5). Note that compared to prior VLMs (e.g., LLaVA) that directly fuse the patch-level features $\mathbf{v}$ of the whole image (Eq. 6) with object information scattering around different positions in $\mathbf{v}$, our object encoding is computationally more efficient and speeds up the training that involves multiple in-context objects in the multimodal prompt.

4 Experimental Settings

In this section, we first describe two main object-level tasks for evaluation (§4.1) together with the datasets used (§4.2). Finally, we describe three variants of our model (§4.3), the training details §4.4), and the other baselines in comparison (§4.5).

4.1 Object-level Tasks

Referring Object Classification Given an object referred by its image location (e.g. segmentation mask/bounding box), the model is instructed to generate a text that predicts the object's class label in a predefined label set, $\mathcal{C} \in \{c_1, c_2, ...c_n\}$. We provide the ground truth segmentation mask to eliminate localization errors and focus on evaluating the models' understanding of image objects.

Referring Expression Generation Given an input image object referred by a segmentation mask, the model is instructed to generate a natural language expression which semantically matches multiple ground-truth references $\mathcal{R} \in \{r_1, r_2, ...r_m\}$. We use METEOR (Banerjee and Lavie, 2005) and CIDEr (Vedantam et al., 2015) score for evaluating generated description quality.

4.2 Datasets

This section describes the different datasets used in our experiments, with more details in Appendix C.

Common Objects in Context (COCO) (Lin et al., 2014) is a popular visual reasoning dataset with over 800,000 object-level annotations for 80 categories of objects. We use it to train our model to understand region input since it contains high-quality segmentation annotations. We use the standardized train and validation 2017 splits for the detection task, and discard a few (<1%) small segmentation annotations that fail to be converted into a binary mask. Following (Zhong et al., 2022), we evaluate in the setting where ground-truth segmentations are provided as input to eliminate localization errors. We use the standard metric of mean average precision (mAP) for object detection using the COCO API,[2] as well as overall accuracy.

refCOCOg (Kazemzadeh et al., 2014) is a variant of the COCO dataset with about 50,000 annotations for objects and their description. We use the data to train our model to describe image regions and use their standardized train/validation split.

ChestX-Ray8 (CXR8) (Wang et al., 2017) is a medical dataset consisting of 108,948 frontal-view X-ray images. The image annotations for the 8 possible pathologies are text-mined from the radiology reports using NLP tools. A small subset of 984 images contains bounding box annotation of the pathology. We use this subset for our zero-shot domain adaptation experiments, splitting the data into 16% retrieval set and 84% test data. The retrieval set consists of 20 examples of each pathology, and we use overall accuracy as the evaluation metric.

[2] https://github.com/cocodataset/cocoapi

4.3 OLIVE Variants

OLIVE-R (Retrieval-only) This retrieval-only method predicts the answer to the user question by taking a majority vote of the top k retrieved examples. For simplicity, we fix $k = 5$ for this setting unless otherwise specified and analyze the effect of k in Figure 6. Although simple, this baseline proves to be effective and provides salient additional context as described in §4.3. However, this discriminative model does not allow for free-form text generation for tasks such as region captioning.

OLIVE-G (Generative-only) This model is trained to generate free-form text based solely on the user question and corresponding object features. We omit the retrieved information to observe the capability of the standalone object representations. We find that even without retrieval, the model can learn to perform more challenging object-level tasks such as region description. The final decoder input can be expressed as a variant of Eq. (10):

$$\mathbf{x} = \texttt{Prompt}(x, \mathbf{v}_{\text{obj}}). \tag{11}$$

OLIVE-RG (Full) Our full model generates text outputs based on in-context object examples from retrieval. The multimodal in-context prompt is constructed using Eq. (10). This prompt includes the retrieved object features, their labels, and their similarity scores. The exact construction can be found in Appendix A. The top k retrieved multimodal documents in $\mathcal{K}$ are obtained using the same retrieval described in §3.2 and ordered in increasing relevance score. Both OLIVE-G and OLIVE-RG use greedy decoding for text generation.

4.4 Training Details

Our model uses a frozen `ViT-L/14` vision transformer from a CLIP model to obtain patch-level features. For our LLM backbone, we use either Llama 2-7B or GPT-2 (124M) (Radford et al., 2019). The LLM is finetuned with LoRA (Hu et al., 2021) as we find this improves model performance. We use the train splits of two different region-level datasets (i.e., COCO, refCOCOg) as our training data for their respective tasks, and evaluate models on their corresponding validation splits because their test data does not have object-level annotation. More details are in Table 7 and we leave the other hyperparameter search to future exploration. We additionally find that we can train a multi-task model by combining the datasets for all object-level tasks (Details in Appendix E).

4.5 Other Baselines in Comparison

CLIP (Radford et al., 2021) Contrastive Language Image Pretraining learns a joint vision-language space between images and their matching captions. We use this method for zero shot object classification by predicting the target with the highest cosine similarity to the cropped region.

BioMedCLIP (Zhang et al., 2023a) The authors train a CLIP model aligned to biomedical image-text pairs, achieving state of the art on a variety of medical tasks. We use this model as a baseline for object classification in the medical domain.

RegionCLIP (Zhong et al., 2022) This model learns region-text level alignment through soft-labels obtained from CLIP. We use it for referring object detection based on ROIAlign features.

Kosmos 2 (Peng et al., 2024) This generative VLM trains a LLM decoder to perform a variety of visual grounding tasks from their newly introduced grounded image-text (GRIT) dataset. We compare with their results on referring expression generation on the refCOCOg dataset.

Flamingo (Alayrac et al., 2022) This generative model learns to connect frozen visual features and LLMs by training on interleaved image-text data. We evaluate Flamingo's few-shot performance on referring expression generation on cropped image regions. We use an open-source implementation trained on the multimodal C4 (Zhu et al., 2023b) and LAION-2b (Schuhmann et al., 2022) datasets.

5 Results and Analysis

Method Type	Pre-Training Data	Method	Accuracy
Classification	None	OLIVE-R	33.5
	PMC-15	BioMedCLIP	32.5
	PMC-15	BioMedCLIP$_{crop}$	23.3
	CLIP400M	CLIP	14.0
	None	Random Guess	12.5
	CLIP400M	CLIP$_{crop}$	11.2
Generative	COCO	OLIVE-RG	31.2
	C4 + LAION-2b	Flamingo-9B	12.5
	COCO	OLIVE-G	0.0

Table 2: Zero shot transfer results of our Llama 2 backbone referring object classification model to new objects. *crop* indicates cropping the image based on the bounding box annotation and using it as the input image. Flamingo is evaluated 8-shot (one example from each pathology).

5.1 Referring Object Classification

Unseen Object Classification One of the benefits of our retrieval augmented system is its rapid

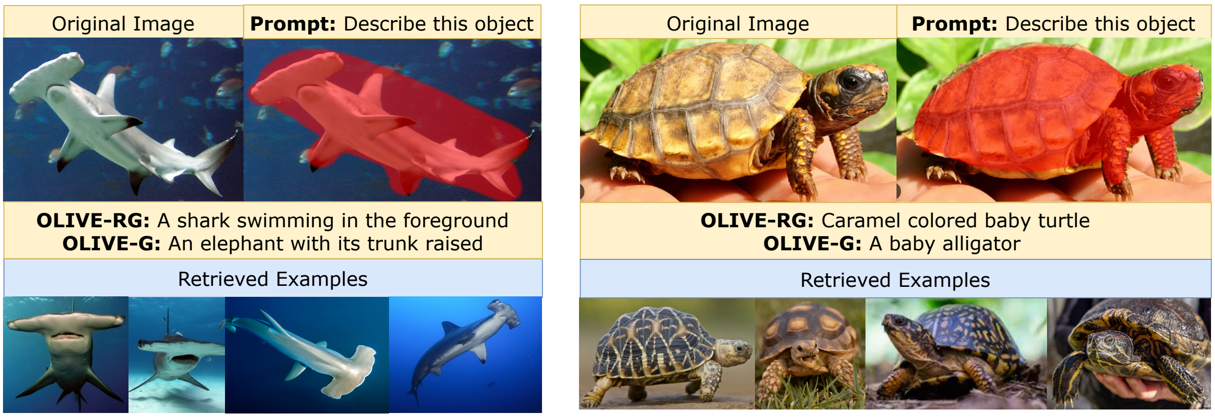

Original Image
Prompt: Describe this object
OLIVE-RG: A shark swimming in the foreground
OLIVE-G: An elephant with its trunk raised
Retrieved Examples
Original Image
Prompt: Describe this object
OLIVE-RG: Caramel colored baby turtle
OLIVE-G: A baby alligator
Retrieved Examples

Figure 2: Examples showcasing the benefit of using retrieval for out of distribution objects. Despite not being trained with any images of sharks or turtles, OLIVE-RG can describe them zero shot by adding a few pictures of them in the retrieval set.

generalization to unseen visual concepts. We estimate this capability by training on the COCO dataset and evaluating object classification on an unseen medical dataset which has drastically different types of images and limited training data. Table 2 illustrates the performance of our method on the CXR8 dataset in either a classification or generative setting. Even with as little as 20 examples per class in the medical retrieval set, OLIVE-R achieves competitive performance compared to domain-adapted models (i.e., BioMedCLIP), which we hypothesize is because of our region-level retrieval and in-distribution retrieval set. We also note that our generative approach OLIVE-RG can utilize the retrieved in-context examples and achieve similar performance to BioMedCLIP, despite only being trained on COCO images. Without retrieval, the generative model fails catastrophically with 0% accuracy, and zero-shot CLIP achieves about the same performance as random guessing.

Rare Object Classification We also investigate our model's performance on rare, but seen objects. Figure 3 shows our method's performance on the top 5 rarest classes in the COCO dataset. For OLIVE-G and OLIVE-RG, we use a 224 pixel resolution visual encoder to match the CLIP visual encoder. OLIVE-G tends to have lower performances on the rare classes. However, when combining retrieval with parameterized methods in OLIVE-RG and OLIVE-RG-336px, the performance on rare classes improves significantly, with OLIVE-RG-336px performing better than CLIP on all rare classes. OLIVE-RG also achieves better performance on three out of five classes despite being trained on less data. Our model's overall performance can be found in Appendix D.

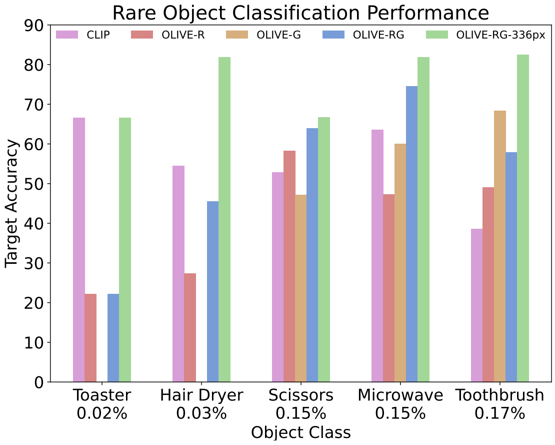
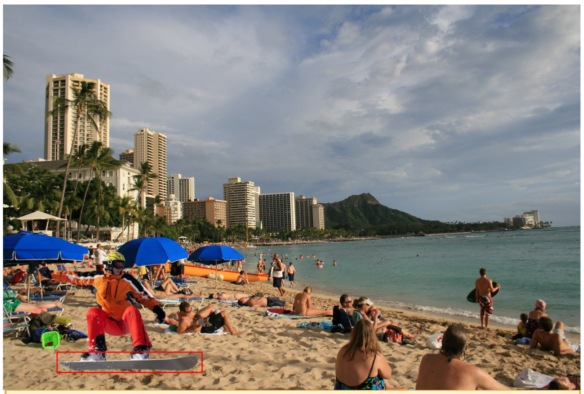


Figure 3: Classification accuracy on the most uncommon objects in the COCO dataset. We use a Llama 2 decoder backbone and numbers below the object classes indicate their proportion of the retrieval set. Combining retrieval with generative methods improves performance and increasing vision encoder resolution provides further gains.

5.2 Referring Expression Generation

Captioning Unseen Objects In addition to referring object classification, we investigate our model's ability to caption out-of-distribution objects. Figure 2 illustrates an example of asking our model to describe animals not seen during training. Without retrieval, OLIVE-G fails to describe the shark and turtle. However, after manually adding just 5 labeled objects of turtles and sharks to the existing retrieval set, OLIVE-RG accurately describes the object and provides supporting examples for its prediction. The label description for each object in the retrieval set is only the name of the animal, but the model generates additional characteristics in its description. Appendix B shows more examples of zero-shot adaptation to unseen visual concepts in the object classification setting.



Prompt: What is the object in the red rectangle?
Kosmos 2: It is a boogie board
ViP-LLaVA: The object is a skateboard
OLIVE-G (ours): The snowboard of the man

Figure 4: A challenging visual example in which the background of the image does not correspond with the query object. Methods which cross-attend to the whole image struggle to identify the snowboard, while our object representation enables more accurate description.

Challenging Visual Context To test the quality of the representations generated from our object encoder, we qualitatively evaluate our model prediction in adversarial visual contexts. Figure 7 shows a white dog and a black cat in a "yin-yang" shape. We observe that free-form annotation allows for more precise user queries and object descriptions, and illustrates other properties such as scene content awareness and patch-level details as shown in Appendix B. While many VLMs can accurately understand normal scenes, Figure 4 illustrates an example in which an object-level representation may be necessary, with recent works struggling to caption the snowboarder on the beach. The detailed performance of our model on the refCOCOg captioning task can be found in Appendix F.

5.3 In-context Example Size

Since our method omits image patch features and compresses object information into a single vector, it can process many objects from different images at once. In Figure 5, we highlight the difference in context length for various methods when prompted with multimedia examples. We assume an average prompt length of 30 accompanying each in-context image example for all models. Even approaches designed for interleaved image-text data such as Flamingo insert multiple latent vectors for each image, incurring a higher cost than our approach.

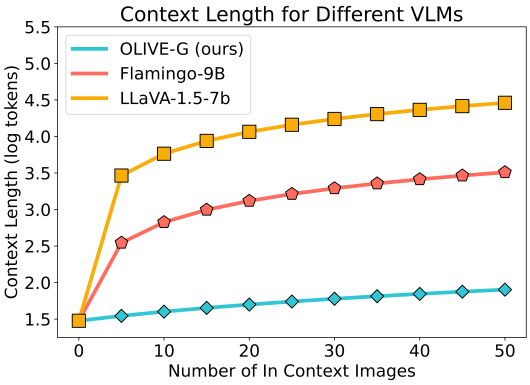


Figure 5: Context length of different VLMs when prompted with multimodal input. Models that represent images with many patch tokens or learned latents incur higher costs with more in-context examples.

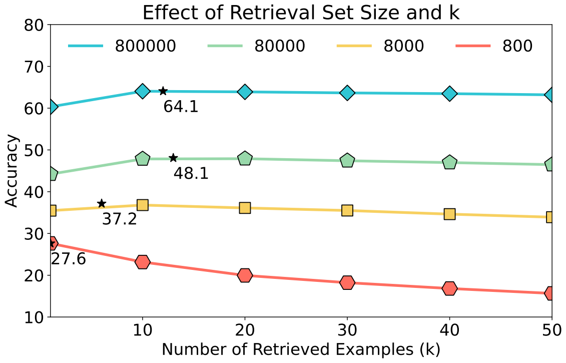


Figure 6: An illustration of how retrieval set size and choice of k affects referring object classification performance. Numbers in the legend indicate the size of the retrieval set, and the stars are the highest accuracies achieved on their curves.

5.4 Sensitivity on Retrieval: Coverage and k

In Figure 6 we analyze the effect of changing the size of the object retrieval set as well as the number of retrieved examples, k. To thoroughly test various settings, we evaluate the retrieval-only based approach (OLIVE-R) on the validation split of the COCO dataset using different sized subsets of the training data for retrieval. We ensure the retrieval set contains an equal amount of each object class when possible. Our results indicate that the optimal value of k depends on the size of the retrieval set. With a small retrieval dataset (red), performance is lower and the optimal k tends to be smaller. Larger retrieval sets (blue, green) benefit from retrieving more examples and have greater performance.

5.5 Object Vector Visualization

Having a single vector representation for each object allows for visualization using dimensionality reduction. In Figure 8, we perform principal com-

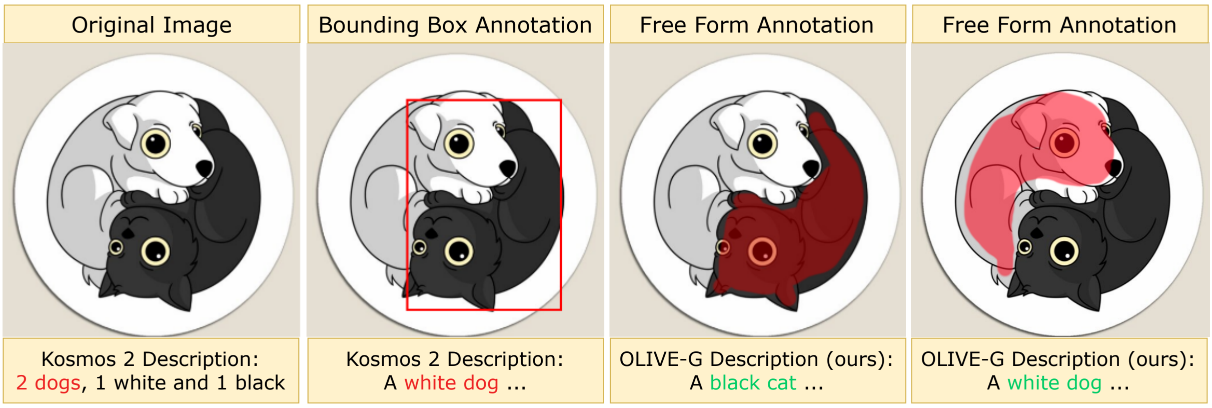


Figure 7: An illustration of the benefit of free-form visual input. Models that use the entire image or bounding box to refer to regions fail to describe the black cat, while OLIVE-G can use free-form annotation to identify the white dog and black cat.

ponent analysis (PCA) on the hidden states of object vectors at different layers in the LLM decoder. We plot 200 examples from each of 10 object categories and note several patterns. First, objects from the same class tend to appear together, even though they appear in different visual contexts. This suggests that the object encoder has semantic understanding of the visual concepts. Second, the object vectors naturally form hierarchical clusters where objects from the same super class such as vehicle, animal, or fruit have overlapping clusters. Lastly, the clustering appears similar across all layers, with only minor variations.

6 Related Work

Grounding in Language and Vision A popular approach for aligning vision and language embeddings is contrastive learning methods such as CLIP and ALIGN (Li et al., 2021). However, these methods align the entire image representation, leading to poor reasoning on image details for downstream vision language tasks. RegionCLIP (Zhong et al., 2022) and GLIP (Li et al., 2022) address this issue by proposing fine-grained alignment with region-text pairs during pretraining. GLIPv2 (Zhang et al., 2022) further improves the pretraining and alignment by introducing localization, detection, and other tasks. Another recent popular approach involves training models on automatically curated region-level data from image-caption pairs (Peng et al., 2024). Many other works focus on region-level alignment during pretraining for greater vision-language understanding (You et al., 2023; Chen et al., 2023; Zeng et al., 2022b,a). More generally, a recent study (Bugliarello et al., 2023) shows that VLMs with fine-grained object-level pretraining such as X-VLM (Zeng et al., 2022a) have better reasoning ability. Other works align vision and language using regularization or loss to create relation aware cross attention between modalities (Pandey et al., 2023; Ren et al., 2021).

Visual Resampling Visual resampling is a popular technique to compress long sequences of image features into a few rich vector representations. This is achieved by constructing a fixed amount of learnable vectors that attend to the visual features through cross-attention layers. Models such as BLIP (Li et al., 2023) first explore this idea to connect frozen vision features to LLMs efficiently by summarizing the content of the image. Other methods including X-Decoder (Zou et al., 2023a) or SEEM (Zou et al., 2023b) use resampling to encode various types of prompts or intents which improve the LLM decoding ability. Additionally, works such as Flamingo (Alayrac et al., 2022) and Qwen-VL (Bai et al., 2023) show that multiple images can be inserted in-context to the prompt by compressing image features with resamplers, enabling few shot capabilities. Our work visually resamples object representations for object-conditioned text generation, and only uses a single vector for the representation. This allows for more fine-grained reasoning and longer in-context prompting.

Retrieval Augmented VLMs In the text domain, learning to retrieve relevant documents to enhance the LLM query (Guu et al., 2020) has been explored extensively (Wang et al., 2023). Recent VLM works follow a similar approach to retrieve multimodal documents to improve performance on knowledge-intensive tasks and improve generalization to rare situations. Gao et al. (2022) summarizes visual content into natural language to use as

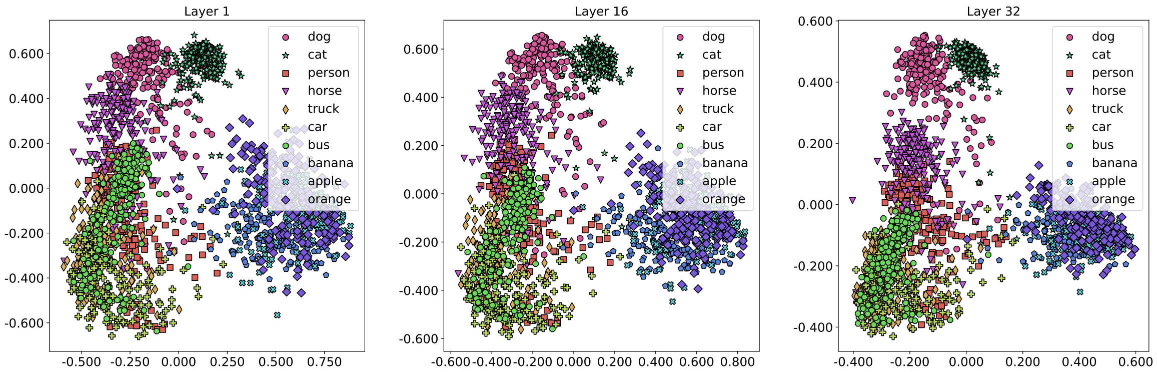


Figure 8: Visualization of the top 2 components when performing PCA decomposition on object vectors in different layers of the LLM decoder (Llama 2). We display 200 examples from each class. Across different layers, objects with similar semantics appear together in the plot, even though they appear in different visual contexts. The clusters are similar across all layers.

a query for dense passage retrieval. MuRAG (Chen et al., 2022) proposes a multimodal image-text memory bank to help models answer challenging knowledge-based visual questions such as “What shape is the pediment on top of the white house?" REVEAL (Hu et al., 2023) and RA-VQA (Lin and Byrne, 2022) learns a trainable multimodal retriever similar to REALM (Guu et al., 2020) during pretraining to fetch relevant documents to answer questions, achieving state of the art performance on datasets such as VQAv2 (Antol et al., 2015) and OKVQA (Schwenk et al., 2022). To the best of our knowledge, we are the first to integrate region-level retrieval with LLMs, in which the multimodal documents are indexed by object-level visual features.

7 Conclusion

We present a simple approach to insert object level visual embeddings into large language model decoders, enabling object level reasoning with flexible prompt structure. Our object encoder compresses fine-grained region level information into a single vector, enabling in-context prompting with objects from multiple images and more efficient training and inference. In addition, we introduce the idea of region retrieval, which allows for precise queries free of image background noise and rapid generalization to rare and unseen objects with no parameter updates. We hope our method may help researchers design vision language models which can adapt to their needs by simply updating the retrieval set or object encoder, while also being responsive to varying user intents using LLM prompting techniques.

8 Limitations

While our approach provides a flexible way for users to supply object-level prompts, it does not output bounding boxes or other region-level grounding. This may be addressed in future research by further finetuning on region-level instruction tuning data as done in FERRET (You et al., 2023), GLAMM (Rasheed et al., 2023), and other region-level VLM pretraining. At the moment, we also do not explore generic image tasks such as VQA or image captioning. However, a potential solution is to use our object encoder to connect to existing VLMs (e.g. LLaVA) which excel at these tasks. Lastly, our results in the retrieval setting depend on the quality of the retrieved examples. Curating a high-quality retrieval set at the object-level can be challenging. However, existing tools such as GLIPv2 (Zhang et al., 2022) allows for semi-automatic generation of region-level data as used in KOSMOS-2 (Peng et al., 2024) in developing the GRIT dataset.

9 Ethical Considerations

Biases From Pretrained LLMs Since our model uses existing pretrained LLMs such as Llama 2 or GPT2, it may inherent some of the social biases or toxicity acquired during their pretraining stages. While Llama 2 undergoes extensive alignment to ideal human values through reinforcement learning from human feedback (RLHF) (Griffith et al., 2013), some of these toxic behaviors may still be present in the morally aligned model. We make sure to only use images of common objects in the COCO dataset, which do not contain any of these biases or violent scenes to the best of our knowledge. Nevertheless, further testing to ensure the impartiality of the model may be necessary before deploying in widespread technologies.

Domain Adaptation Some of our experiments involve evaluating our model in a data-scarce domain in a zero shot manner with in-context prompting. While this is a promising direction for efficient domain adaptation, users should take caution in directly using model prediction, as this is a challenging task due to distribution shift. We encourage human-in-the-loop interaction to sanity check the outputs. Different from other ICL prompting methods, we provide retrieved examples and similarity scores which can help determine the trustworthiness of the model prediction, which may be valuable for high-risk domains such as medicine.

Acknowledgement

Ossowski and Hu are supported by the Wisconsin Alumni Research Foundation and the National Institute Of Biomedical Imaging And Bioengineering of the National Institutes of Health under Award Number R01EB033782. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

References

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. 2022. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35:23716–23736.

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. 2015. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

Emanuele Bugliarello, Laurent Sartran, Aishwarya Agrawal, Lisa Anne Hendricks, and Aida Nematzadeh. 2023. Measuring progress in fine-grained vision-and-language understanding. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*.

Mu Cai, Haotian Liu, Siva Karthik Mustikovela, Gregory P Meyer, Yuning Chai, Dennis Park, and Yong Jae Lee. 2023. Making large multimodal models understand arbitrary visual prompts. *arXiv preprint arXiv:2312.00784*.

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

Wenhu Chen, Hexiang Hu, Xi Chen, Pat Verga, and William W Cohen. 2022. Murag: Multimodal retrieval-augmented generator for open question answering over images and text. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Feng Gao, Qing Ping, Govind Thattai, Aishwarya Reganti, Ying Nian Wu, and Prem Natarajan. 2022. Transform-retrieve-generate: Natural language-centric outside-knowledge visual question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5067–5077.

Shane Griffith, Kaushik Subramanian, Jonathan Scholz, Charles L Isbell, and Andrea L Thomaz. 2013. Policy shaping: Integrating human feedback with reinforcement learning. *Advances in neural information processing systems*, 26.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *International Conference on Machine Learning*, pages 3929–3938. PMLR.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. 2017. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Ziniu Hu, Ahmet Iscen, Chen Sun, Zirui Wang, Kai-Wei Chang, Yizhou Sun, Cordelia Schmid, David A

Ross, and Alireza Fathi. 2023. Reveal: Retrieval-augmented visual-language pre-training with multi-source multimodal knowledge memory. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23369–23379.

Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. 2014. Referitgame: Referring to objects in photographs of natural scenes. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 787–798.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. 2023. Segment anything. *arXiv preprint arXiv:2304.02643*.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International Conference on Machine Learning*.

Junnan Li, Ramprasaath Selvaraju, Akhilesh Gotmare, Shafiq Joty, Caiming Xiong, and Steven Chu Hong Hoi. 2021. Align before fuse: Vision and language representation learning with momentum distillation. *Advances in neural information processing systems*, 34:9694–9705.

Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, et al. 2022. Grounded language-image pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10965–10975.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer.

Weizhe Lin and Bill Byrne. 2022. Retrieval augmented visual question answering with outside knowledge. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. *arXiv preprint arXiv:2304.08485*.

Rohan Pandey, Rulin Shao, Paul Pu Liang, Ruslan Salakhutdinov, and Louis-Philippe Morency. 2023. Cross-modal attention congruence regularization for vision-language relation alignment. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5444–5455, Toronto, Canada. Association for Computational Linguistics.

Zhiliang Peng, Wenhui Wang, Li Dong, Yaru Hao, Shaohan Huang, Shuming Ma, and Furu Wei. 2024. Kosmos-2: Grounding multimodal large language models to the world. In *Proceedings of the Twelfth International Conference on Learning Representations (ICLR)*.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Hanoona Rasheed, Muhammad Maaz, Sahal Shaji, Abdelrahman Shaker, Salman Khan, Hisham Cholakkal, Rao M Anwer, Erix Xing, Ming-Hsuan Yang, and Fahad S Khan. 2023. Glamm: Pixel grounding large multimodal model. *arXiv preprint arXiv:2311.03356*.

Shuhuai Ren, Junyang Lin, Guangxiang Zhao, Rui Men, An Yang, Jingren Zhou, Xu Sun, and Hongxia Yang. 2021. Learning relation alignment for calibrated cross-modal retrieval. *arXiv preprint arXiv:2105.13868*.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. 2022. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35:25278–25294.

Dustin Schwenk, Apoorv Khandelwal, Christopher Clark, Kenneth Marino, and Roozbeh Mottaghi. 2022. A-okvqa: A benchmark for visual question answering using world knowledge. In *European Conference on Computer Vision*, pages 146–162. Springer.

Senthujan Senkaiahliyan, Augustin Toma, Jun Ma, An-Wen Chan, Andrew Ha, Kevin R An, Hrishikesh Suresh, Barry Rubin, and Bo Wang. 2023. Gpt-4v (ision) unsuitable for clinical care and education: A clinician-evaluated assessment. *medRxiv*, pages 2023–11.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575.

Liang Wang, Nan Yang, and Furu Wei. 2023. Learning to retrieve in-context examples for large language models. *arXiv preprint arXiv:2307.07164*.

Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald M Summers. 2017. Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2097–2106.

Jialian Wu, Jianfeng Wang, Zhengyuan Yang, Zhe Gan, Zicheng Liu, Junsong Yuan, and Lijuan Wang. 2022. Grit: A generative region-to-text transformer for object understanding. *arXiv preprint arXiv:2212.00280*.

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. 2023. mplug-owl: Modularization empowers large language models with multimodality.

Haoxuan You, Haotian Zhang, Zhe Gan, Xianzhi Du, Bowen Zhang, Zirui Wang, Liangliang Cao, Shih-Fu Chang, and Yinfei Yang. 2023. Ferret: Refer and ground anything anywhere at any granularity. *arXiv preprint arXiv:2310.07704*.

Licheng Yu, Hao Tan, Mohit Bansal, and Tamara L Berg. 2017. A joint speaker-listener-reinforcer model for referring expressions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7282–7290.

Alireza Zareian, Kevin Dela Rosa, Derek Hao Hu, and Shih-Fu Chang. 2021. Open-vocabulary object detection using captions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14393–14402.

Yan Zeng, Xinsong Zhang, and Hang Li. 2022a. Multi-grained vision language pre-training: Aligning texts with visual concepts. In *Proceedings of the Thirty-ninth International Conference on Machine Learning*.

Yan Zeng, Xinsong Zhang, Hang Li, Jiawei Wang, Jipeng Zhang, and Wangchunshu Zhou. 2022b. x^2-vlm: All-in-one pre-trained model for vision-language tasks. *arXiv preprint arXiv:2211.12402*.

Haotian Zhang, Pengchuan Zhang, Xiaowei Hu, Yen-Chun Chen, Liunian Li, Xiyang Dai, Lijuan Wang, Lu Yuan, Jenq-Neng Hwang, and Jianfeng Gao. 2022. Glipv2: Unifying localization and vision-language understanding. *Advances in Neural Information Processing Systems*, 35:36067–36080.

Sheng Zhang, Yanbo Xu, Naoto Usuyama, Jaspreet Bagga, Robert Tinn, Sam Preston, Rajesh Rao, Mu Wei, Naveen Valluri, Cliff Wong, et al. 2023a. Large-scale domain-specific pretraining for biomedical vision-language processing. *arXiv preprint arXiv:2303.00915*.

Shilong Zhang, Peize Sun, Shoufa Chen, Min Xiao, Wenqi Shao, Wenwei Zhang, Kai Chen, and Ping Luo. 2023b. Gpt4roi: Instruction tuning large language model on region-of-interest. *arXiv preprint arXiv:2307.03601*.

Yiwu Zhong, Jianwei Yang, Pengchuan Zhang, Chunyuan Li, Noel Codella, Liunian Harold Li, Luowei Zhou, Xiyang Dai, Lu Yuan, Yin Li, et al. 2022. Regionclip: Region-based language-image pretraining. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16793–16803.

Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. 2023a. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*.

Wanrong Zhu, Jack Hessel, Anas Awadalla, Samir Yitzhak Gadre, Jesse Dodge, Alex Fang, Youngjae Yu, Ludwig Schmidt, William Yang Wang, and Yejin Choi. 2023b. Multimodal c4: An open, billion-scale corpus of images interleaved with text. In *Advances in Neural Information Processing Systems (D&B)*.

Xueyan Zou, Zi-Yi Dou, Jianwei Yang, Zhe Gan, Linjie Li, Chunyuan Li, Xiyang Dai, Harkirat Behl, Jianfeng Wang, Lu Yuan, et al. 2023a. Generalized decoding for pixel, image, and language. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15116–15127.

Xueyan Zou, Jianwei Yang, Hao Zhang, Feng Li, Linjie Li, Jianfeng Gao, and Yong Jae Lee. 2023b. Segment everything everywhere all at once. In *Advances in Neural Information Processing Systems (poster)*.

A Appendix

Table 3 contains all the prompts we use to instruct the LLM decoder.

B Qualitative Examples

Here we include several selected examples showcasing the strengths and weaknesses of our approach.

Visual Concept Generalization In Figure 9 we demonstrate more examples of rapid generalization to new visual concepts. Many existing methods confidently predict concepts from their pretraining, while ours can predict new concepts on the fly.

Scene Content Awareness Even though our object representation involves masking out image patch features from other parts of the image, we have observed that the object vector still contains information about its surroundings. Figure 10 illustrates this phenomenon, where OLIVE can include the cow in its description, despite not including any image patches corresponding to the cow in the user selection.

Patch level Detail Our method also can identify and describe small objects at the patch level. Figure 11 shows an example of object classification on smaller objects.

Describing Partially Visible Objects We notice that our model can make mistakes when describing occluded or partially visible objects as seen in Figure 12. We hypothesize that the training data of refCOCOg does not include these kinds of image regions, which also limits its availability in retrieval data. This may be addressed with larger-scale pretraining on data such as GRIT which likely includes more occluded objects.

Errors in Detailed Description While our model can identify the object most of the time, it sometimes gets minor details incorrect. For example, the colors of a shirt or other piece of clothing are seen in Figure 12. This may be due to the extreme compression we learn into a single vector. Future work may consider visually resampling the object features into more than just 1 latent vector for detailed captioning, but still use the single vector representation for retrieval.

C Dataset Information

Table 4 provides more details on the dataset splits used in our training and evaluation. Our COCO train and validation splits are slightly smaller than normal because of our approach of using segmentation masks. We decide to omit some excessively small segmentations which account for less than 1% of the data. For tasks that require training (COCO and refCOCOg), we use the train split of the COCO object detection dataset as our retrieval data. We make sure to omit the closest match when training object detection on COCO with retrieval to avoid label leakage. We also confirm that no images are repeated in the validation split from the training split for both datasets.

D Referring Object Classification

This task requires the LLM to predict the object class label given a ground truth input annotation (e.g. bounding box, segmentation, etc). We follow a similar evaluation protocol used in (Zhong et al., 2022) and (Zareian et al., 2021), in which the ground truth annotation is supplied to avoid localization error. Table 5 shows the overall referring object classification accuracy and mAP [3] for our methods. We observe several findings. First, although retrieved examples help with domain adaption and rare objects, it does not improve the overall in-domain performance. Second, both the LLama 2 and GPT2 baseline have similar performances on the task, suggesting that even smaller models can learn vision-language grounding. Lastly, even our retrieval-only baseline, which requires no training, has better accuracy than some parameterized methods such as CLIP.

E Multi-Task Model

We also explore the possibility of training a multi-task model using a similar curriculum learning strategy to LLaVA (Liu et al., 2023). We first train the model on the referring object classification task to perform the object-word level alignment. The model is then trained on the referring expression generation task, and finally on an object instruction following dataset (Cai et al., 2023) with many different tasks. For each stage of training, we formulate the task in an instruction-following manner through the prompts in Table 3. This allows the

[3] To simplify the calculation, we assigned a confidence score of 1 to each prediction. Reported mAP may be lower than the true value when using more accurate probabilities.

Task	ICL Prompt for Retrieved Examples	Vanilla Prompts
Object Classification	You are a helpful vision assistant trained to help people analyze images. The top [k] related objects are: [obj] is a [label] with confidence [score] [obj] is a [label] with confidence [score] ⋮ [vanilla prompt]	[obj] What is this? Answer in 1-2 words [obj] What is this object? Answer with a short word or phrase. [obj] Identify this object. Here is an object [obj]. What is this? Answer with a short word or phrase.
Region Description	You are a helpful vision assistant trained to help people analyze images. The top [k] related objects are: [obj] is a [label] with confidence [score] [obj] is a [label] with confidence [score] ⋮ [vanilla prompt]	[obj] Briefly describe this image region. [obj] Describe this part of the image. [obj] Share some details about about what's happening here in the image. [obj] Break down what you see in this particular part of the picture. [obj] Describe what you notice in this area of the picture.

Table 3: Collection of the prompts we use to guide the LLM decoder generation. For a given task, we sample one of the vanilla prompts uniformly at random. Text in brackets indicates variables whose value is dynamically filled in.

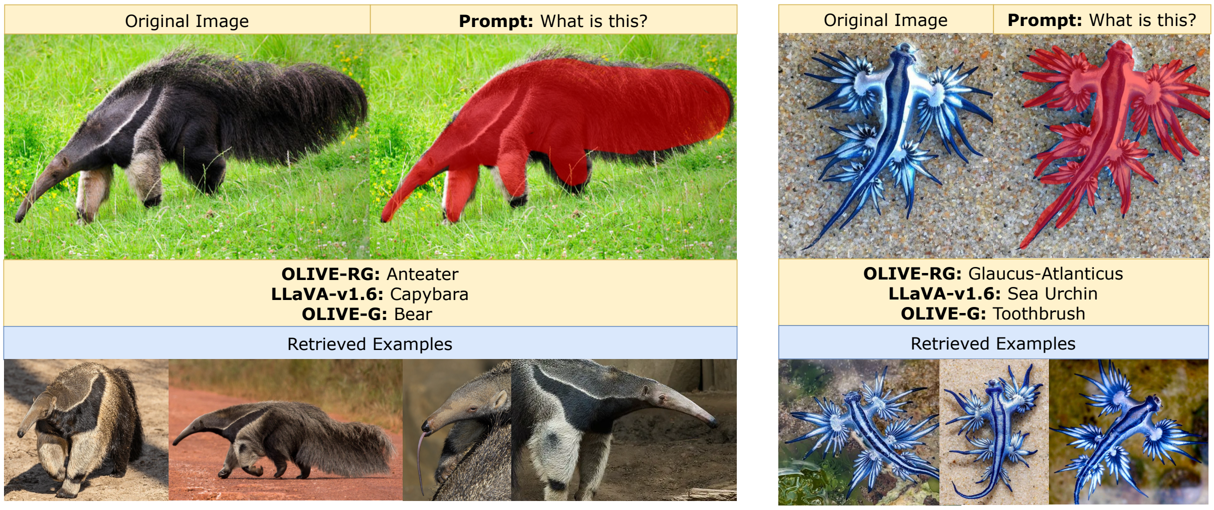


Figure 9: Examples of rapid adaptation to unseen visual concepts during training. Non-retrieval-based methods such as LLaVA often fail to generalize, instead predicting animals seen during pre-training.

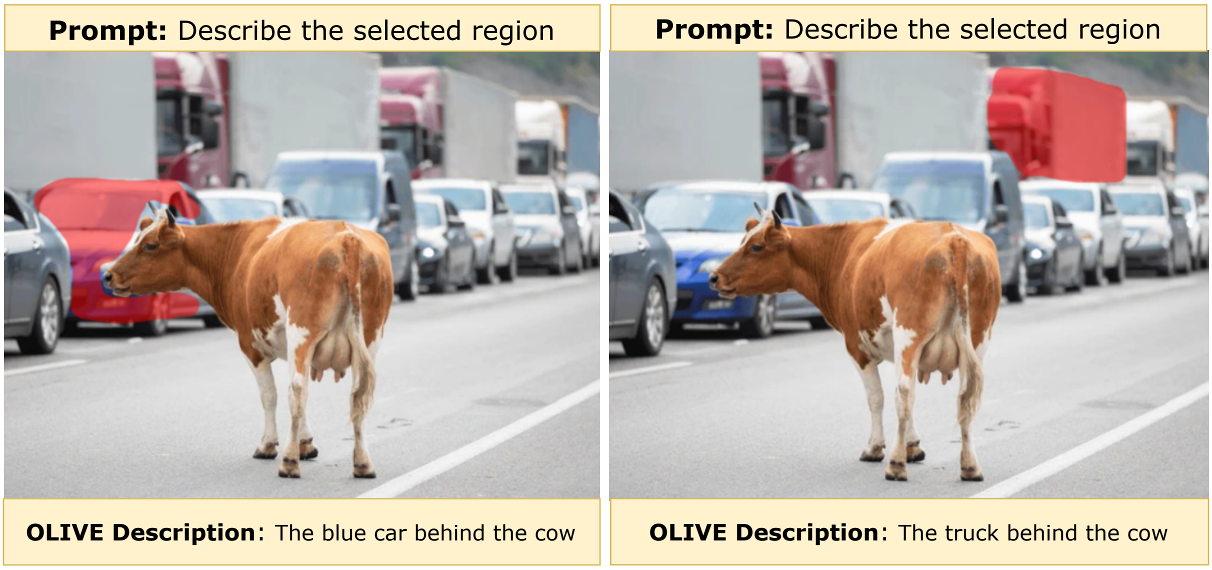

Prompt: Describe the selected region
OLIVE Description: The blue car behind the cow
Prompt: Describe the selected region
OLIVE Description: The truck behind the cow

Figure 10: Example of scene content awareness of the object embeddings. Although neither selection includes any part of the cow, the model can still mention the cow in its description.

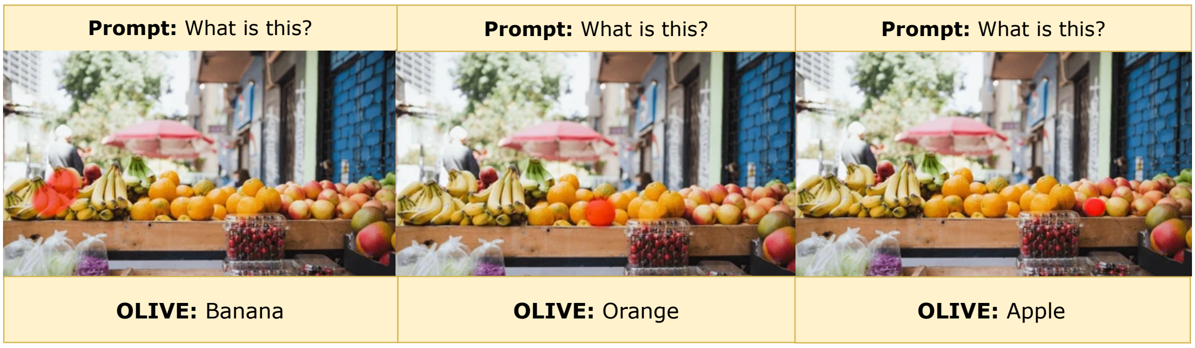


Figure 11: Object classification on small objects.

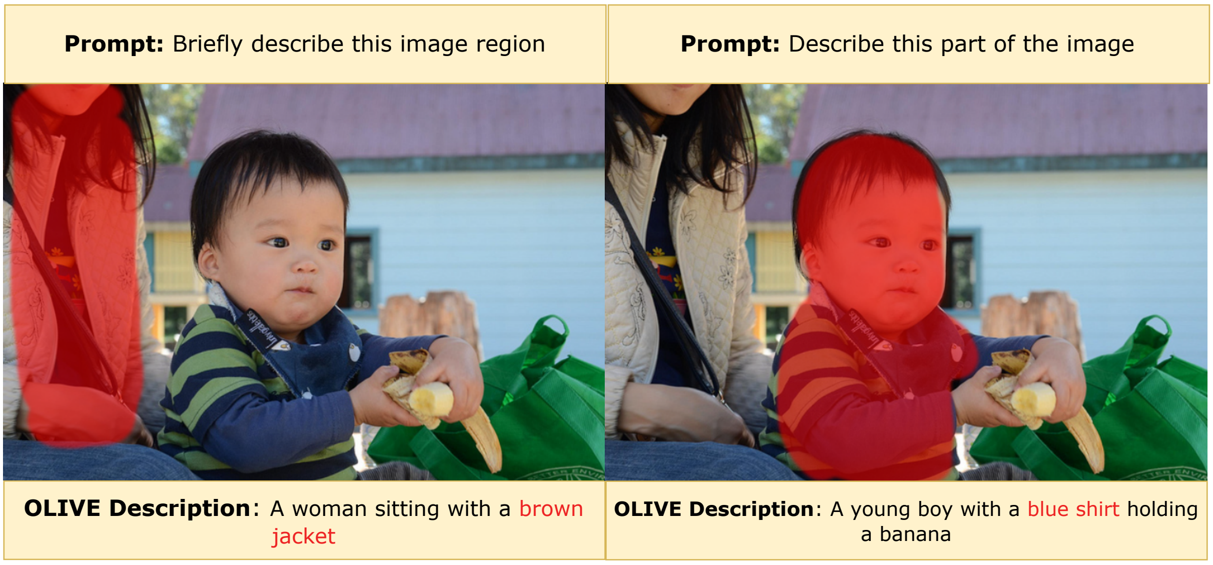
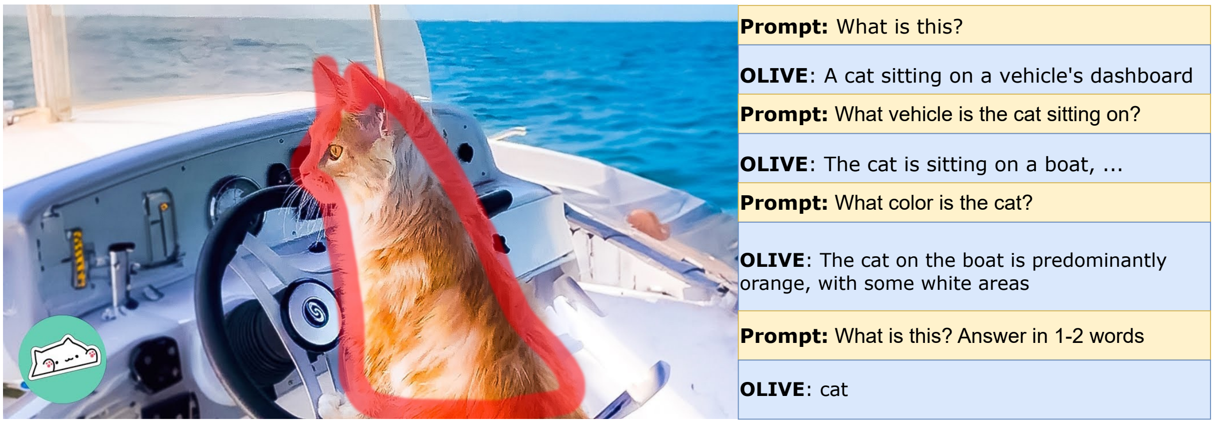

Prompt: Briefly describe this image region
Prompt: Describe this part of the image
OLIVE Description: A woman sitting with a brown jacket
OLIVE Description: A young boy with a blue shirt holding a banana

Figure 12: Failure cases of our model. The model sometimes struggles with occluded/partially missing objects and may mix up some fine-grained details about objects.

Dataset	Train Split	Validation Split	Retrieval Set Train Split	Retrieval Set Test Split	Number of Classes
COCO	849586	36320	849586	849586	80
refCOCOg	44822	5000	849586	849586	-
CXR8	-	824	-	160	8

Table 4: More details about the datasets and splits we use in our experiments. For COCO and refCOCOg, we use the train split of the COCO dataset as the retrieval data. We select the first 20 examples of each pathology as the retrieval set for the zero shot evaluation on CXR8.



Prompt: What is this?

OLIVE: A cat sitting on a vehicle's dashboard

Prompt: What vehicle is the cat sitting on?

OLIVE: The cat is sitting on a boat, ...

Prompt: What color is the cat?

OLIVE: The cat on the boat is predominantly orange, with some white areas

Prompt: What is this? Answer in 1-2 words

OLIVE: cat

Figure 13: When trained on multiple tasks and object instruction following data, OLIVE is able to respond to varying user intents.

Method Type	Method	Accuracy	mAP
Classification	OLIVE-R	**64.1**	40.5
	CLIP `ViT-L/14`	40.9	45.1
	RegionCLIP `RN50`	-	**61.4**
	OVR	-	44.5
Generative	OLIVE-G (GPT2)	76.6	**60.4**
	OLIVE-G (Llama 2)	**76.8**	60.3
	OLIVE-RG (GPT2)	74.8	57.5
	OLIVE-RG (Llama 2)	74.1	56.2

Table 5: Performances on the referring object classification task with different levels of context on the COCO dataset. For each method type, the highest values for each metric are bolded.

model to be responsive to many different user intents (Figure 13)

F Referring Expression Generation

Method	METEOR	CIDEr
OLIVE-G (Llama 2)	16.5	64.0
OLIVE-RG (Llama 2)	16.6	67.7
OLIVE-G (GPT2)	16.4	70.9
OLIVE-RG (GPT2)	**17.0**	75.0
SLR (Yu et al., 2017)	15.4	59.2
SLR+Rerank (Yu et al., 2017)	15.9	66.2
GLAMM (Rasheed et al., 2023)	16.2	**105.0**
GRIT (Wu et al., 2022)	15.2	71.6
Kosmos 2 (zero shot)	12.2	60.3
Kosmos 2 (fewshot k = 2)	13.8	62.2
Kosmos 2 (fewshot k = 4)	14.1	62.2
Flamingo-9B (zero shot)	9.2	34.3
Flamingo-9B (fewshot k = 2)	10.2	36.2
Flamingo-9B (fewshot k = 4)	12.3	39.6

Table 6: Referring expression generation on the refCOCOg validation split. Our approach has competitive perfomance compared to other notable methods which also offer multi-modal in-context prompting.

We study our model's overall performance on referring expression generation by quantitatively evaluating our model on the RefCOCOg validation set shown in Table 6. Several findings can be observed. First, including retrieved multimodal documents results in slightly better performance. Second, the size of the LLM can be modified without much performance change, with GPT2 performing slightly better than Llama 2. Third, having global image context contained in the object representation is important, as methods that crop the image region (e.g. Flamingo) perform worse.

G Training Hyperparameters

We provide the detailed training hyperparameters in Table 7.

Hyperparameter	Classification	Generation
Epochs	1	5
Batch Size	4	4
Training Steps	$\sim$ 200,000	$\sim$ 56,030
Learning Rate	2e-5	2e-5
Optimizer	Adam	Adam
GPU Used	GTX 3090	GTX 3090
Train Time (hours)	24	7.5

Table 7: Details about the hyperparameters we use for (1) Referring Object Classification (**Classification**) and (2) Referring Expression Generation (**Generation**) in our experiments.