Reasoning to Attend: Try to Understand How <SEG> Token Works

Rui Qian[1] Xin Yin[2] Dejing Dou[1*]
[1]School of Computer Science and Technology, Fudan University
[2]The State Key Laboratory of Blockchain and Data Security, Zhejiang University
qianruii@126.com, xyin@zju.edu.cn, dejingdou@gmail.com

Abstract

*Current Large Multimodal Models (LMMs) empowered visual grounding typically rely on <SEG> token as a text prompt to jointly optimize the vision-language model (e.g., LLaVA) and the downstream task-specified model (e.g., SAM). However, we observe that little research has looked into how it works. In this work, we first visualize the similarity maps, which are obtained by computing the semantic similarity between the <SEG> token and the image token embeddings derived from the last hidden layer in both the LLaVA encoder and SAM decoder. Intriguingly, we have found that a striking consistency holds in terms of activation responses in the similarity map, which reveals that what <SEG> token contributes to is the semantic similarity within image-text pairs. Specifically, <SEG> token, a placeholder expanded in text vocabulary, extensively queries among individual tokenized image patches to match the semantics of an object from text to the paired image while the Large Language Models (LLMs) are being fine-tuned. Upon the above findings, we present READ, which facilitates LMMs' resilient **REA**soning capability of where to atten**D** under the guidance of highly activated points borrowed from similarity maps. Remarkably, READ features an intuitive design, Similarity as Points module (SasP), which can be seamlessly applied to <SEG>-like paradigms in a plug-and-play fashion. Also, extensive experiments have been conducted on the ReasonSeg and RefCOCO(+/g) datasets. To validate whether READ suffers from catastrophic forgetting of previous skills after fine-tuning, we further assess its generation ability on an augmented FP-RefCOCO(+/g) dataset. All codes and models are publicly available at https://github.com/rui-qian/READ.*

1. Introduction

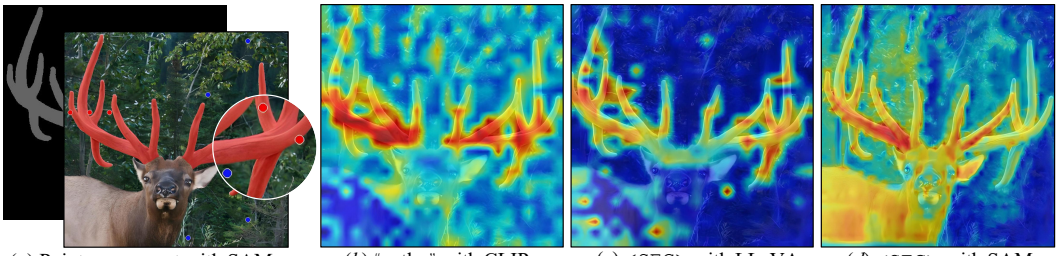
Reasoning segmentation has been newly proposed yet largely unexplored at present [15]. As an extension of classical Referring Expression Segmentation (RES) [10], it aims to output nuanced masks for implicitly referred objects given descriptive language expressions. As shown in Fig. 1, when asked, "What part of the deer's body in the picture is used for defense and attracting mates?", reasoning segmentation infers "antler" without an explicit mention, in contrast to traditional RES, which relies on direct referring. Solving such intricate visual tasks is non-trivial, requiring models to comprehend user intentions based on given queries while also possessing pertinent world knowledge [42].

Recent works [15, 32, 35, 42, 43] have advanced reason segmentation tasks by leveraging <SEG> token as a text prompt to seamlessly align LMMs empowered visual encoder (*e.g.*, LLaVA [21]) and the downstream task-specified decoder (*e.g.*, SAM [14]) in vision space for fine-grained output formats, *i.e.*, segmentation masks. Specifically, SESAME [15] teaches LMMs to respond to false premises by introducing negative samples into the pipeline. GSVA [43] bridges the gap where the multiple-target and empty-target cases are neglected. GLaMM [32] and PixelLM [35] enhances the model both in textual and visual domains, with versatile capability at various levels of granularity.

However, we observe that little research has looked into how <SEG> token works when mapping language vocabulary embedding into corresponding visual space. The <SEG> token, an extended placeholder in the text vocabulary, lacks inherent semantics on its own. However, when it was inserted into conversation templates and jointly trained with LMMs, ultimately being able to ground objects within an image. Recent works [15, 42, 43] all employ the SAM [14] model as a mask decoder. Initially, segmenting the red region in Fig. 1 could be achieved by prompting SAM with the text "antler," but now <SEG> token embedding fulfills the same purpose in place. This leads us to ask: *What is the connection of embeddings between <SEG> token and text prompt "antler" in terms of semantics?*

Bearing this in mind, we begin by visualizing similarity maps, which are generated by computing dot product similarity between <SEG> token and image token embeddings extracted from the last hidden layer in both LLaVA [21] and SAM [14] models. Notably, we observe a striking consistency in activation responses across this similarity maps,

*Corresponding Author (dejingdou@gmail.com).



(a) Points as prompt with SAM (b) "antler" with CLIP (c) <SEG> with LLaVA (d) <SEG> with SAM

Figure 1. Qualitative analysis of <SEG> token on ReasonSeg *train* split. Points derived from (c) serve as prompt with original SAM in (a). Text "antler" with image token from CLIP is in (b). The similarity between <SEG> token and image token embeddings stemming from the last hidden layer are obtained by Eq.(5), w.r.t. LLaVA encoder in (c) and SAM decoder in (d). The consistency in (b), (c), (d) indicates that <SEG> token in LMMs learns semantics similar to those of direct mention in texts. Refer to Appendix B for more illustrations.

suggesting that <SEG> token is pivotal in bridging semantic connections between textual prompts and their visual correspondences. Specifically, <SEG> token, an expansion of text vocabulary, initiates a thorough query across each tokenized image patch, aligning the textual semantics of an object with its visual counterparts in the paired image. Inspired by the above findings, an intuitive idea is to see if we can imply to the model where to "attend" by leveraging similarity maps.

To this end, we present READ, which facilitates LMMs' resilient **REA**soning capabilities of where to atten**D**, informed by highly activated points stemming from similarity maps. In particular, our READ consists of three core modules: (1) a LLaVA encoder, (2) a Similarity as Points module (SasP), and (3) a SAM decoder. Specifically, our LLaVA [21] enhanced encoder consumes image-text pairs as input to generate text output, from which the last hidden layer embedding for <SEG> token is then gathered. To guide the model where to attend, our SasP module computes similarity maps by performing a dot product between the <SEG> token embedding and their associated image patches, whereupon regions with high similarity scores are then converted into point coordinates for fine-grained mask predictions through the SAM decoder, along with textual prompts, *i.e.* the <SEG> embedding. To address the challenge posed by discrete, non-differentiable points during back-propagation, we apply a Gaussian-like weighted average interpolation to render them continuously differentiable. This modification facilitates gradients across similarity maps back to the LMMs, empowering the model to "reason to attend" in the forwards, and "attend to reason" in the backward and vice versa. Particularly, READ's intuitive design, SasP, can be effortlessly integrated into off-the-shelf <SEG>-like paradigms with minimal overheads in a plug-and-play manner. In summary, our contributions are threefold:

- We have looked into how <SEG> token works when mapping language vocabulary embedding into corresponding visual space. Such investigation reveals that what <SEG> token mainly contributes to is the semantic correspondences from image-text pairs based on our findings.
- We present our model — READ, which empowers LMMs to "reason to attend" and "attend to reason" vice versa. Importantly, our intuitive and simple design, SasP, can be effortlessly integrated into off-the-shelf <SEG>-like pipelines with marginal overheads.
- We conduct extensive experiments on both the challenging reasoning segmentation dataset and the well-established RefCOCO(+/g) referring segmentation dataset. To see if READ struggles with catastrophic forgetting of previous capabilities, we also assess its generative performance on the FP-RefCOCO(+/g) dataset. Our READ excels existing state-of-the-arts a by remarkable margin, resulting in cIoU improvements over baselines of up to 4.7% on ReasonSeg, 3.73% on FP-RefCOCO(+/g).

2. Related Work

2.1. Large Multimodal Models

Depending on the level of capabilities LMMs possess, we categorize them into three different groups: (1) *Multimodal feature alignment*, which aligns visual and textual features toward a comprehensive multimodal understanding [1, 17, 46]. Flamingo [1] enables visual in-context reasoning through cross-attention. BLIP-2 [17] utilizes a lightweight query-based visual encoder to align image features with a frozen language model. mPLUG-OWL [46] connects image encoders to language models via efficient prompts. (2) *Instruction tuning for few-shot learning*, which leverages instruction tuning to enable few-shot learning capabilities, allowing models to obtain new skills even with limited samples [16, 21, 49]. Otter [16] enhances LMMs with instruction-tuning on the proposed MIMIC-IT dataset. LLaVA [21] and MiniGPT-4 [49] combine a visual encoder for feature extraction and align with text embeddings, effectively boosting its capabilities across a range of vision-language tasks. (3) *Fused task and enhanced reasoning*, which advances LMMs' reasoning capabilities with a unified interface for versatile vision-centric tasks [28, 29, 39, 47]. VisionLLM [39] unifies diverse visual tasks within a single language model interface. Kosmos-2 [28] and DetGPT [29] inject grounding capabilities into LMMs. GPT4RoI [47] allows region-level understanding by constructing region-text pairs. In contrast, our READ builds upon LLaVA [21] for world knowledge and complex reasoning.

2.2. Interactive Segmentation Models

Depending on whether the interactive capabilities are present in the models, we archive literature into two groups: (1) *Non-interactive segmentation*, which assigns class labels to pixel-level in a scene (*i.e.*, semantic segmentation [2, 24, 36]), or object-level of a scene (*i.e.*, instance segmentation [4, 7, 22]), or both of which simultaneously (*i.e.*, panoptic segmentation [12, 13, 44]). U-Net [36] features an encoder-decoder architecture that incorporates skip connections to preserve feature information. Mask R-CNN [7] introduces an additional segmentation branch to Faster R-CNN [34], allowing for object detection and instance segmentation in parallel. PS [13] initially introduces the concept of panoptic segmentation, defining it as the task of labeling every pixel in an image, including both segmentable objects (like people and vehicles, *etc*.) and unsegmentable 'stuff' classes (such as sky and road, *etc*.). (2) *Interactive segmentation*, which aims to interact with human language, enabling models to segment target objects based on descriptive texts, typically, Referring Expression Segmentation (RES) [15, 32, 35, 42, 43]. GSVA [43] boosts RES by enabling the identification of multiple objects from a single description and recognizing absent targets. GLaMM [32] and PixelLM [35] augment models' capabilities in both textual and visual domains, showcasing versatility across multiple levels of granularity. Closest to our work, LISA [15] and SESAME [42] inject self-reasoning capabilities into segmentation tasks, elevating RES into even more advanced interactions, *i.e.*, reasoning segmentation.

Note that aforementioned literature, w.r.t. RES leans on <SEG> token as the intermediate connector to link the downstream mask decoder. Whereas, we observe that few investigations have looked into how it works so far. This work aims to unveil how <SEG> token contributes to, whereupon we present the proposed method, READ.

3. Reflection on Reasoning Segmentation

In this section, we first revisit the reasoning segmentation task and then analyze the underlying mechanisms of how <SEG> token works upon the prior state-of-the-art [15, 42].

3.1. Revisiting

Problem Definition: Let $\mathbf{x}_{img} \in \mathbb{R}^{h\times w\times c}$ denote the input image, where h, w and c are height, width, and channel of the image, respectively. Consider the paired textual input, denoted by $\mathbf{x}_{txt}$, which could range from an explicit mention, such as "antler" to an implicit expression like "part of the deer's body". Reasoning segmentation primarily involves the task of generating a segmentation mask $\hat{\mathbf{M}}$ such that it aligns with the part of the image that corresponds to the referenced query as

$$\Theta_{MLE} = \arg\max_{\Theta} \mathcal{G}_\theta \left(\hat{\mathbf{M}}|\mathbf{x}_{img}, \mathbf{x}_{txt}; \Theta\right), \tag{1}$$

where $\mathcal{G}_\theta$ indicates segmentation capabilities infused LMMs parameterized by Θ. In this paper, it includes a multi-modal LLM $\mathcal{G}_\mathcal{T}$ and a visual backbone model $\mathcal{G}_\mathcal{V}$. Concisely, $\mathcal{G}_\theta = \mathcal{G}_\mathcal{T} \oplus \mathcal{G}_\mathcal{V}$, $\oplus$ denotes cascading operation. As illustrated in Fig. 2, $\mathcal{G}_\mathcal{T}$ and $\mathcal{G}_\mathcal{V}$ are instantiated by LLaVA [21] and SAM [14] accordingly. $\hat{\mathbf{M}} \in \{0, 1\}^{h\times w}$, 1 indicates the presence of the object and 0 otherwise.

To facilitate $\mathcal{G}_\theta$ in a embedding as mask fashion, LISA [15] extends the text vocabulary of $\mathcal{G}_\mathcal{T}$ with a placeholder, *i.e.*, <SEG> token. To allow LLMs to tackle image features in the same way as text sequences, $\mathbf{x}_{img}$ is segmented into patches of equal size and transformed by the CLIP [30] model. During training, <SEG> token is embedded within $\mathbf{x}_{txt}$, and then both $\mathbf{x}_{txt}$ and $\mathbf{x}_{img}$ are fed into LMMs $\mathcal{G}_\mathcal{T}$, which in turn generates a text responses $\hat{\mathbf{y}}_{txt}$ as

$$\hat{\mathbf{y}}_{txt} = \mathcal{G}_\mathcal{T}(\mathbf{x}_{img}, \mathbf{x}_{txt}). \tag{2}$$

During inference, when $\mathcal{G}_\theta$ is prompted interactively to output a binary segmentation mask, the response $\hat{\mathbf{y}}_{txt}$ would include a <SEG> token if the object exists. LISA [15] then retrieves the last hidden layer embedding $\tilde{\boldsymbol{h}}_{seg}$ from $\mathcal{G}_\mathcal{T}$ associated with the predicted <SEG> token, which then is passed through a multilayer perceptron (MLP) projection layer, denoted by φ to obtain the refined feature $\boldsymbol{h}_{seg}$. Concurrently, the vision backbone $\mathcal{G}_\mathcal{V}^{enc}$ borrowed from SAM [14] is employed to extract a rich set of visual features $\mathbf{f}$ from the visual input $\mathbf{x}_{img}$, represented by

$$\boldsymbol{h}_{seg} = \varphi(\tilde{\boldsymbol{h}}_{seg}), \quad \mathbf{f} = \mathcal{G}_\mathcal{V}^{enc}(\mathbf{x}_{img}). \tag{3}$$

At last, visual features $\mathbf{f}$ are passed into mask decoder $\mathcal{G}_\mathcal{V}^{dec}$ to produce the final segmentation mask $\hat{\mathbf{M}}$ conditioned by embedding $\boldsymbol{h}_{seg}$ as

$$\hat{\mathbf{M}} = \mathcal{G}_\mathcal{V}^{dec}(\mathbf{f}, \boldsymbol{h}_{seg}). \tag{4}$$

$\boldsymbol{h}_{seg}$ importantly bridges the intermediate layer, informing the mask decoder to seamlessly decode the segmentation mask in a way of end-to-end training. This paradigm has been widely adopted by its successors [32, 35, 42, 43]. However, we observe that few investigations have looked into how the embedding $\boldsymbol{h}_{seg}$ works so far, which inspires us to explore its underlying mechanism.

3.2. Analysis

To qualitatively analyze the <SEG> token, we visualize the similarity maps at different stages of the forward pass through $\mathcal{G}_\theta$ in SESAME [42]. As illustrated in Fig. 1, "<SEG> with LLaVA" denotes the similarity map between the embedding of <SEG> token, *i.e.*, $\boldsymbol{h}_{seg}$ and the image tokens from the last hidden layer outputs. Activation responses in (c) and (d) indicate that $\boldsymbol{h}_{seg}$ signifies the model where to "attend" somehow. Further, the consistency between the

striking clues in (b) and (c) reveals that $\boldsymbol{h}_{seg}$ plays a key role in aligning with the semantics when bridging the intermediate layers. (b) is obtained by using the embedding of text "antler" and visual features from CLIP [30], which indirectly suggests that the <SEG> token has acquired semantics similar to those of the direct mention "antler" when prompted by the implicit expression "part of the deer's body".

To quantitatively assess the effects of <SEG> token, we conduct experiments, w.r.t. similarity maps. We first select several points with the highest similarity scores as positives and an equal number of points with the lowest similarity scores as negatives. These points are then directly used as prompts, rather than the <SEG> token, and input into the original SAM model to generate the segmentation mask (see Appendix 5). In Table 1, $\mathcal{P}_{\text{prompt}}$ reveals that only relying on the selected similarity points can still generate a segmentation mask potentially (27.0%vs.30.4%). To further illustrate to what extent the activated points within the similarity map in (c) correspond to the target object in (a), we adopt the grid search-based Intersection over Union (see Appendix A.1) to validate the consistency between the similarity map and the ground-truth mask. In Table 1, $\mathcal{S}_{\text{IoU}}$ suggests that responses in the similarity map are strikingly consistent with the ground-truth mask, with a 6% higher cIoU on ReasonSeg *test* split (36.4%vs.30.4%) in the 2^{nd} row.

Table 1. Quantitative analysis of <SEG> token on ReasonSeg *test* split. $\mathcal{P}_{\text{prompt}}$ denotes points as prompt for original SAM [14]. $\mathcal{S}_{\text{IoU}}$ measures the overlap (IoU) between the similarity map and the ground-truth mask. <SEG>$_{\text{prompt}}$ denotes the <SEG> token as prompt for the adapted SAM, a.k.a. LISA [15].

Method	$\mathcal{P}_{\text{prompt}}$		$\mathcal{S}_{\text{IoU}}$		<SEG>$_{\text{prompt}}$	
	gIoU	cIoU	gIoU	cIoU	gIoU	cIoU
LISA-7B [15]	38.2	30.1	32.1	34.1	47.3	48.4
SESAME [42]	35.9	27.0	32.5	36.4	30.5	30.4

Summary. In summary, by analyzing the effects of <SEG> token, we observe the following: (a) The <SEG> token in LMMs learns semantic features similar to those of direct mentions in texts and aligns these textual semantics with its visual space to control the generation of segmentation mask. (b) The activated points within similarity maps imply the locations of the target object, which to some extent provides feedback on where the model is focusing. Understanding how the <SEG> token works is crucial as it is closely tied to the issue of semantic alignment within LLMs. <SEG> token offers insights into the experimental observation that, when prompting LISA [15] for reasoning explanations, the textual outputs from the LLaVA encoder remain accurate, even for cases where the SAM decoder fails in segmentation. These reflections motivate us to directly leverage similarity points to guide the model where to "attend" when reasoning. For more analysis, please see the Appendix B.

4. Proposed READ

In this section, we present READ, which unlocks LMMs' resilient reasoning capability of where to "attend" under the guidance of highly activated points derived from similarity maps. As shown in Fig. 2, the proposed READ includes: (a) an LLaVA encoder, (b) a Similarity as Points Prompter (SasP), and (c) a SAM decoder. In READ, we first use the LLaVA encoder to take image-text pairs as input, which in turn responds to text as outputs. We then extract the embedding of <SEG> token and image tokens from the last hidden layer of the LLaVA encoder to compute the similarity map, upon which we employ our Discrete to Continuous sampling (DtoC) to convert highly activated foreground points into continuous ones. Finally, these continuous points along with <SEG> token embedding are fed into the SAM [14] decoder for mask generation. As these points are differentiable, loss will be backpropagated to LMMs to signify $\mathcal{G}_{\mathcal{T}}$ "where to attend" when reasoning. Given that our innovation mainly lies in SasP, we discuss it first in Sec. 4.1.

In what follows, we present the LLaVA encoder in Sec. 4.2. SAM [14] decoder in Sec. 4.3, and the Training objectives of READ are detailed in Sec. 4.4.

4.1. Similarity as Points

Points as Prompt. Inspired by various types of prompts supported by SAM [14] mask decoder, such as bounding boxes, points, and dense mask, *etc*., we explore how to derive the points of interests as prompt in the input image $\mathbf{x}_{img}$ via the similarity score. Specifically, denote $\boldsymbol{h}^{(l_k)} = \left\{\boldsymbol{h}_1^{(l_k)}, ..., \boldsymbol{h}_{N_k}^{(l_k)} | \boldsymbol{h}_i^{(l_k)} \in \mathbb{R}^d\right\}$ as the hidden state output at the k-th layer of $\mathcal{G}_{\mathcal{T}}$, where N_k denotes the number of hidden state tokens, and d is the embedding dimension. $\boldsymbol{h}_{seg}$ in Eq. (3) can be represented as $\boldsymbol{h}_{seg}^{(l_k)} \in \boldsymbol{h}^{(l_k)}$. In the course of training, the image features are incorporated into the text instruction $\mathbf{x}_{txt}$ and fed as input to the LLM. Hence, the hidden state output at the k-th layer, $\boldsymbol{h}^{(l_k)}$, includes N_t image tokens, denoted as $\boldsymbol{h}_{img}^{(l_k)} \subseteq \boldsymbol{h}^{(l_k)}$. We formulate the similarity score between <SEG> token and image tokens as

$$\mathcal{S} = \boldsymbol{h}_{img}^{(l_k)} \cdot \left(\boldsymbol{h}_{seg}^{(l_k)}\right)^{\mathrm{T}}, \tag{5}$$

where $\mathcal{S}$ denotes similarity score between each image token and the <SEG> token, $\mathcal{S} \in \mathbb{R}^{N_t}$. Note that our vanilla similarity score is parameter-free, which could be easily amenable to <SEG>-like paradigms with negligible efforts. Since this paper primarily aims to explore how <SEG> token works, we leave designing possibly more effective similarity computation strategy for our READ as future work, such as using cross attention [38] to acquire learnable fine-grained patterns. Nevertheless, Eq. (5) is already sufficient enough to generate the necessary clues according to (c) in Fig. 1.

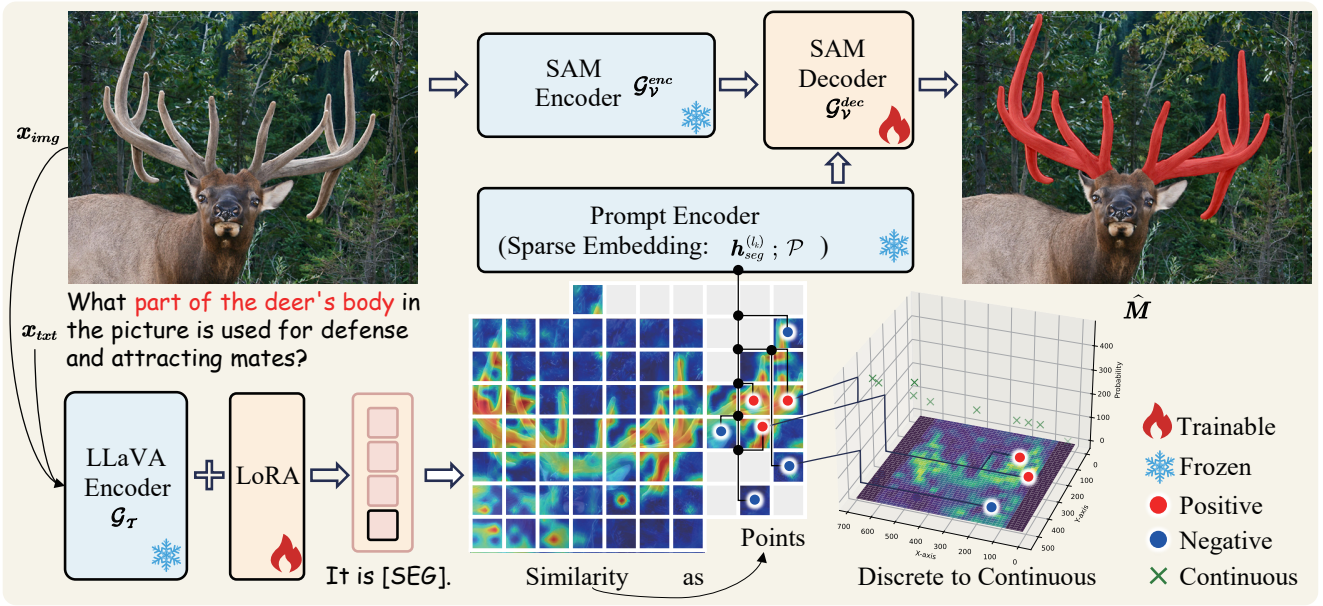


Figure 2. Overview of our proposed READ. The hidden state outputs with respect to the <SEG> token and image tokens are derived from the LLaVA encoder for similarity as points, before being fed into the prompt encoder for sparse embedding. To inform the model where to "attend" when reasoning, we apply a Gaussian-like weighted average interpolation to transform discrete points into continuous ones.

To facilitate similarity score as point prompt, we restore the coordinates over three types of points in $\mathbf{x}_{img}$, *i.e.*, positive, negative, and neutral points. Positive points denote those that are confidently associated with an object or the foreground. Negative points denote those that are confidently associated with the background. Neutral points denote those that are not clearly identifiable as belonging to either the object or the background, which could be near object boundaries, ambiguous regions, or areas. To this end, Algorithm (1) outlines the procedure: (i) In Steps 1-3, we normalize the similarity score $\mathcal{S}$ and set the thresholds for positive and negative points based on the mean μ and variance σ of $\mathcal{S}$. (ii) In Steps 6-7, we recovery each selected point j with the absolute coordinates. (iii) In Steps 8-10, since the point selection process involves operations such as sorting, j becomes a discrete, non-differentiable value. To allow for gradient backpropagation, we apply a Gaussian-like weighted average interpolation to obtain continuous, differentiable coordinates. The weights are computed based on the distance to each grid point. After that, point set $\mathcal{P}$ along with <SEG> token embedding $\boldsymbol{h}_{seg}^{(l_k)}$ is fed into $\mathcal{G}_{\mathcal{V}}^{dec}$ as input, we reformulate Eq. (4) as

$$\hat{\mathbf{M}} = \mathcal{G}_{\mathcal{V}}^{dec}(\mathbf{f}, \boldsymbol{h}_{seg}^{(l_k)}, \mathcal{P}). \tag{6}$$

The shift from discrete points to continuous, differentiable ones is crucial, as the gradients propagated backward during training are expected to guide the model in refining its focus. If the similarity score of background points is higher, causing positive points to fall within the background, it will degrade the mask result and increase the corresponding loss, which in turn, penalizes the model and encourages it to learn where to "attend". Considering that Steps 8-10 form the core of SasP, we go through it in detail.

Discrete to Continuous. Let $\mathbf{g}_{\boldsymbol{x}}$ and $\mathbf{g}_{\boldsymbol{y}}$ represent the coordinates of grid points, and (x_j, y_j) denote the coordinates of the selected point j. By using distance-based weights and normalized softmax probabilities, the sampling process will be continuously differentiable with respect to selected point (x_j, y_j) in the limit as the grid resolution increases. Specifically, the weight for each grid point is computed based on the distance to the selected point, using an exponential decay function. This ensures that closer points have higher weights, and farther points have lower weights as

$$w_i^j = \exp\left(-d_i^j\right), \tag{7}$$

where d_i^j denotes the distance between grid point i and the selected point j. To incorporate both the distance weight and the softmax probability p_i (from the similarity map $\mathcal{S}$), the final weight for each grid point is computed as

$$\tilde{w}_i^j = w_i^j \cdot p_i, p_i = \frac{\exp\left(\mathcal{S}_i\right)}{\Sigma_{i'} \exp\left(\mathcal{S}_{i'}\right)}. \tag{8}$$

These weights are normalized so that their sum equals 1, ensuring that they form a valid probability distribution as

$$\hat{w}_i^j = \frac{\tilde{w}_i^j}{\Sigma_{i'} \tilde{w}_{i'}^j}, \tag{9}$$

where the sum is taken over all grid points i'. Finally, the continuous coordinates $(\hat{x}_i, \hat{y}_j)$ for each selected point (x_j, y_j)

are computed by taking a weighted average of the grid coordinates, using the normalized final weights $\hat{w}_i^j$ as

$$\hat{x}_j = \sum_{i=1}^{h\times w} \mathbf{g}_{x,i} \cdot \hat{w}_i^j, \hat{y}_j = \sum_{i=1}^{h\times w} \mathbf{g}_{y,i} \cdot \hat{w}_i^j, \tag{10}$$

where $\mathrm{g}_{x,i}$ and $\mathrm{g}_{y,i}$ are the x-th and y-th coordinates of grid point i, respectively. Given that the weight function $\exp\left(-d_i^j\right)$ decays smoothly with the distance between g_i and (x_j, y_j), $\hat{w}_i^j$ forms a smooth probability distribution over the grid points. Thereby, the weighted sum of grid points converges to continuous interpolation as the grid is refined, *i.e.*, in the limit as the grid resolution $\Delta x \to 0$ (the distance between grid points approaches zero), the discrete weighted sum can be approximated by an integral over a continuous domain. The weighted average becomes

$$\hat{x}_j = \int \mathbf{g}_x\left(\mathbf{g}\right) \cdot w\left(\mathbf{g}, x_j, y_j\right)\ d\mathbf{g}, \tag{11}$$

where $\mathbf{g}$ is a continuous variable representing the position on the grid, and $w\left(\mathbf{g}, x_j, y_j\ \right)$ is a smooth weight function based on the distance to the selected point (x_j, y_j).

4.2. LLaVA Encoder

To encode image and text features in parallel, we follow the approach in works [15, 42] and instantiate $\mathcal{G}_\mathcal{T}$ with LLaVA [21]. Specifically, $\mathcal{G}_\mathcal{T}$ consists of a CLIP [30] model for processing image features and a LLaMA [37] model for processing text features. First, the CLIP model convolves $\mathbf{x}_{img}$ into N_t image patches, each of which is then encoded by a series of stacked vision transformers. Next, these image embeddings are projected via MLPs to the same dimension as the text features and embedded into the text instructions $\mathbf{x}_{txt}$ before being fed into the LLaMA, large language model. Note that the parameters of $\mathcal{G}_\mathcal{T}$ are frozen. We utilize LoRA [9] for efficient fine-tuning as work [15].

4.3. SAM Mask Decoder

To decode the segmentation mask based on the text instructions $\mathbf{x}_{txt}$, we deploy SAM [14] model as the visual backbone [42]. First, the SAM model uses a prompt encoder to project sparse embeddings, w.r.t. $\boldsymbol{h}_{seg}^{(l_k)}, \mathcal{P}$ as queries. Also, the SAM image encoder, composed of a series of vision transformers, encodes visual embeddings of $\mathbf{x}_{img}$ as keys. Next, queries and keys interact through 2 layers of the TwoWayAttention Block before finally being decoded into the mask.

4.4. Training Objectives

To infuse segmentation capabilities into the LMMs $\mathcal{G}_\theta$, we jointly optimize the text generation loss $\mathcal{L}_{txt}$ in $\mathcal{G}_\mathcal{T}$, and the segmentation mask loss $\mathcal{L}_{mask}$ in $\mathcal{G}_\mathcal{V}$ [15, 42]. Specifically,

Algorithm 1 Similarity as Points Algorithm

Require:

$\mathcal{S}$ is the similarity score between each image token and the <SEG> token obtained in Eq. (5), where $\mathcal{S} \in \mathbb{R}^{N_t}$, N_t denotes the number of image tokens;

$\mathcal{I}_+$ is the indices of positive points determined by a threshold t_{pos},$\mathcal{I}_-$ is the indices of negative points with a threshold of t_{neg}, $\mathcal{I}_0$ stands for the indices of neutral points. $\mathcal{I} = \mathcal{I}_+ \cup \mathcal{I}_- \cup \mathcal{I}_0$;

$\mathbf{g}$ is the set of grid points, $\mathbf{g} = [h] \times [w]$, where $[h] = \{1, 2, ..., h\}$ and $[w] = \{1, 2, ..., w\}$. hw denote the raw image $\mathbf{x}_{img}$ dimension. $\mathbf{g}_{\boldsymbol{x}}$ and $\mathbf{g}_{\boldsymbol{y}}$ are the coordinates of grid points along x-axis, y-axis.

Ensure:

Selected points and labels: $\mathcal{P} = \emptyset$, labels $= [-1]^{|\mathcal{I}|}$

$\mathcal{S}_{[0,1]} \leftarrow \mathcal{S}, \mu = \mathbb{E}\left[\mathcal{S}_{[0,1]}\right], \sigma^2 = \mathrm{Var}\left[\mathcal{S}_{[0,1]}\right]$

$t_{pos} = \mu + \sigma \cdot \varepsilon, t_{neg} = \mu - \sigma \cdot \varepsilon, \varepsilon = 0.5$

$\mathcal{I}_+ = \{j \mid \mathcal{S}_j \geq t_{pos}\}, \mathcal{I}_- = \{j \mid \mathcal{S}_j \leq\ t_{neg}\}$

$\alpha = \frac{w}{\lfloor\sqrt{N_t}\rfloor}, \beta = \frac{h}{\lfloor\sqrt{N_t}\rfloor}$

for each index j in $\mathcal{I}$ **do**

  $x_j = \min\left((j \bmod w + 0.5) \cdot \alpha, w - 1\right)$

  $y_j = \min\left((j \div w + 0.5) \cdot \beta, h - 1\right)$

  $d_i^j = \left\|(\mathbf{g}_{x,i}, \mathbf{g}_{y,i}) - (x_j, y_j)\right\|_2, \forall i \in \mathbf{g}$

  $\hat{w}_i^j = \frac{\exp\left(-d_i^j\right) \cdot p_i}{\Sigma_{i'} \exp\left(-d_{i'}^j\right) \cdot p_{i'}},\ p_i = \frac{\exp(\mathcal{S}_i)}{\Sigma_{i'} \exp(\mathcal{S}_{i'})}, \forall i \in \mathbf{g}$

  $\hat{x}_j = \sum_{i=1}^{h\times w} \mathbf{g}_{x,i} \cdot \hat{w}_i^j, \quad \hat{y}_j = \sum_{i=1}^{h\times w} \mathbf{g}_{y,i} \cdot \hat{w}_i^j$

  $\mathcal{P} \leftarrow \mathcal{P} \cup (\hat{x}_j, \hat{y}_j)$

  if $j \in \mathcal{I}_+$, $\mathrm{labels}_j = 1$, if $j \in \mathcal{I}_-$, $\mathrm{labels}_j = 0$

end for

return $\mathcal{P}$, labels

we use cross-entropy loss for $\mathcal{L}_{txt}$, pixel-wise binary cross-entropy (BCE) loss and DICE loss for $\mathcal{L}_{mask}$ as

$$\begin{aligned} \mathcal{L}_{txt} &= \mathcal{L}_{ce}(\hat{\mathbf{y}}_{txt}, \mathbf{y}_{txt}), \\ \mathcal{L}_{mask} &= \lambda_{bce}\mathcal{L}_{bce}(\hat{\mathbf{M}}, \mathbf{M}) + \lambda_{dice}\mathcal{L}_{dice}(\hat{\mathbf{M}}, \mathbf{M}), \end{aligned} \tag{12}$$

where λ_{bce} and λ_{dice} are the loss weights, $\mathbf{y}_{txt}$ and $\mathbf{M}$ are the ground-truth targets. The overall objective $\mathcal{L}$ aggregates those losses in Eq. (12), weighted by λ_{txt} and λ_{mask} as

$$\mathcal{L} = \lambda_{txt}\mathcal{L}_{txt} + \lambda_{mask}\mathcal{L}_{mask}. \tag{13}$$

5. Experiment

5.1. Experimental Setting

Network Architecture. We employ LLaVA 1.5-7B [21] as the base language model $\mathcal{G}_\mathcal{T}$ and the ViT-H SAM [14] as the visual backbone $\mathcal{G}_\mathcal{V}$. For image encoding, we adopt the clip-vit-large-patch14-336, which takes 336×336 pixel images as input. The projection layer γ consists of a series of stacked MLPs with channel dimensions [512, 4096, 4096].

Table 2. Comparisons of the state-of-the-art reasoning segmentation results on ReasonSeg dataset. * means results are reproduced by the official model. 'ft' denotes using 239 reasoning segmentation samples to fine-tune the model.

Method	val		test					
	overall		short query		long query		overall	
	gIoU	cIoU	gIoU	cIoU	gIoU	cIoU	gIoU	cIoU
X-Decoder [50]	22.6	17.9	20.4	11.6	22.2	17.5	21.7	16.3
Grounded-SAM [23]	26.0	14.5	17.8	10.8	22.4	18.6	21.3	16.4
SEEM [51]	25.5	21.2	20.1	11.5	25.6	20.8	24.3	18.7
OVSeg [18]	28.5	18.6	18.0	15.5	28.7	22.5	26.1	20.8
GRES [19]	22.4	19.9	17.6	15.0	22.6	23.8	21.3	22.0
*SESAME [42]	34.8	39.1	28.3	27.6	31.6	32.7	30.5	30.4
LLaVAv1.5-7B + OVSeg [15]	38.2	23.5	24.2	18.7	44.6	37.1	39.7	31.8
LISA-7B [15]	52.9	54.0	40.6	40.6	49.4	51.0	47.3	48.4
LISA-7B-LLaVAv1.5 (ft) [15]	**61.3**	62.9	**48.3**	**46.3**	57.9	59.7	55.6	56.9
READ-7B-LLaVAv1.5 (ft)	59.8	**67.6**	47.0	44.4	**60.0**	**63.1**	**56.8**	**59.0**

Implementation Details. We train on 4 NVIDIA 24GB 3090 GPUs for 20 epochs around 24 hours. We deploy DeepSpeed [33] engine for distributed training, with a batch size per device of 2 and, a gradient accumulation step of 10. The AdamW [25] optimizer is initialized with a learning rate of 0.0003 and no weight decay (set to 0). The learning rate is updated by the WarmupDecayLR scheduler, with 100 warmup iterations. The weights for $\mathcal{L}_{mask}$ and $\mathcal{L}_{txt}$ are set to 1.0, and the BCE loss weight λ_{bce} is set to 2.0 and the DICE loss weight λ_{dice} to 0.5 by default. Each image is randomly assigned up to 3 categories before being decorated with a question-and-answer template.

Datasets. We follow prior work LISA [15] to organize data structure, which typically consists of three types of datasets: (1) As for semantic segmentation dataset, we use ADE20K [48] and COCO-Stuff [3], PACO-LVIS [31], and PASCAL-Part [5]; (2) As for referring segmentation dataset, we use refCLEF, refCOCO, refCOCO+ [11], refCOCOg [27], and ReasonSeg [15]; (3) As for the visual question answering (VQA) dataset, we use LLaVA-Instruct-150k for LLaVA v1.5 [21]. Also, to teach READ to overcome false premises, we include FP-RefCOCO(+/g) [42], R-RefCOCO [41] for false premises assessment.

Evaluation Metrics. We adhere to the practices established in prior works [11, 15, 27] by employing two evaluation metrics: gIoU and cIoU. The gIoU metric is calculated as the mean of the Intersection-over-Union (IoU) values across all individual images, and cIoU is computed by the cumulative intersection across the cumulative union.

5.2. Results on ReasonSeg Dataset

Comparison with State-of-the-Arts. To evaluate the performance of READ on the ReasonSeg dataset, we use *train* split for training and validate the performance on *val* set and *test* set. Table 3 reveals that our READ model achieves significant performance gains in both gIoU and cIoU scores, particularly in the more challenging long query scenarios. Specifically, the READ-7B outperforms LISA-7B-LLaVAv1.5 on the ReasonSeg dataset, with a 4.7% higher cIoU on the val set (67.6%vs.62.9%) and a 2.1% improvement in the overall test set (59.0%vs.56.9%). Also, the READ-13B shows a 0.6% advantage (62.8%vs.62.2%) , while in the short query subset, it has a 3.1% lead (53.7%vs.50.6%). Overall, READ consistently surpasses LISA-7B(13B)-LLaVAv1.5 by up to 4.7% across different subsets. This validates the effectiveness of our approach in handling complex reasoning tasks and generating accurate segmentation masks.

Table 3. Comparisons of the state-of-the-art results on ReasonSeg *test* split using LLaVA 1.5-13B [20] as the base language model.

Method	short query		long query		overall	
	gIoU	cIoU	gIoU	cIoU	gIoU	cIoU
LISA-13B-LLaVA1.5 (ft) [15]	55.4	50.6	63.2	**65.3**	61.3	62.2
READ-13B-LLaVAv1.5 (ft)	**55.4**	**53.7**	**64.4**	65.1	**62.2**	**62.8**

5.3. Results on RefCOCO(+/g) Dataset

Comparison with State-of-the-Arts. To demonstrate the efficacy of the proposed READ in the referring segmentation task, we conduct a comparative analysis with the existing state-of-the-art method, as detailed in Table 4. Our evaluation encompasses the refCOCO, refCOCO+, and refCOCOg *val* split and *test* split. The outcomes reveal that our model outperforms existing methods across various referring segmentation tasks. Specifically, READ achieves 3.2% higher cIoU on refCOCO val (78.1%vs.74.9%) and 2.4% higher on refCOCO+ val set (68.4%vs.66.0%). On the refCOCOg set, READ shows an advantage of 2.2% higher on val(U) (70.1%vs.67.9%) and 0.8% higher on test(U) (71.4%vs.70.6%). Overall, READ consistently performs better or on par than LISA-7B across all subsets.

5.4. Results on FP-RefCOCO(+/g) Dataset

Comparison with State-of-the-Arts. To validate whether READ can overcome false premises, we assess its generation ability on augmented FP-RefCOCO(+/g) dataset. Given that LISA [15] is trained on positive samples only, in a fashion which always encourages the model to output mask, even if the object described in the query does not actually exist. As a result, LISA suffers from catastrophic forgetting of previous skills after fine-tuning [42]. We follow SESAME [42], using see scores for the binary classification accuracy and cIoU

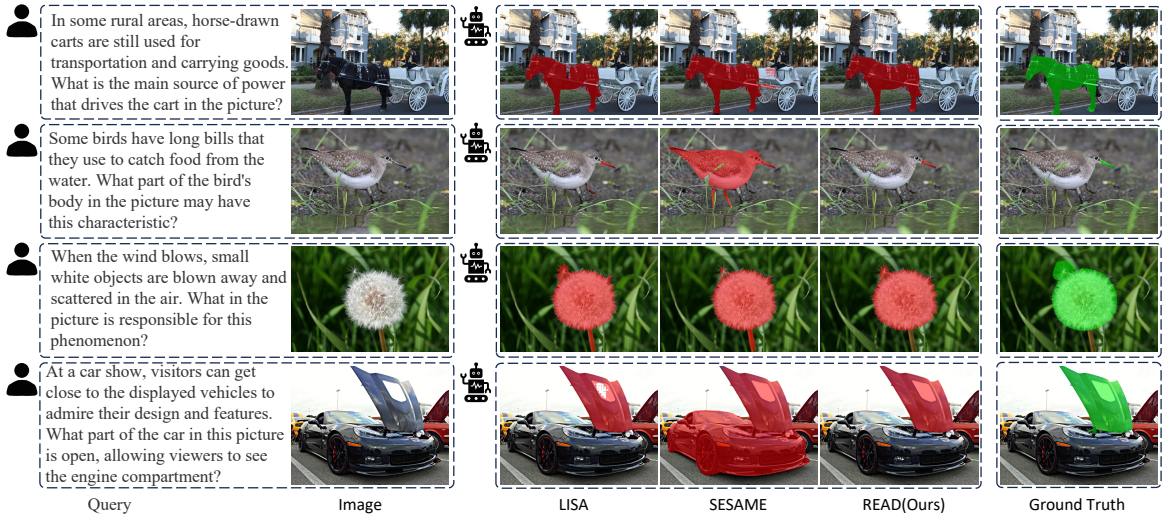


Figure 3. Visual comparison among READ (ours) and prior works on ReasonSeg *val* split. Refer to Appendix D for more illustrations.

for the segment capability. Table 5 shows that READ excels SESAME [42] in the "see" task across all datasets. Specifically, READ improves by 3.03% on FP-RefCOCO, 3.51% on FP-RefCOCO+, and 2.89% on FP-RefCOCOg. For the "segment" task, READ edges out SESAME, with a 3.57% advantage on FP-RefCOCO (61.50%vs.57.93%), 3.73% on FP-RefCOCO+ (54.54%vs.50.81%), and 2.33% on FP-RefCOCOg (56.12%vs.53.79%). This indicates READ's capability in detecting the existence of objects while generating segmentation masks even under false premises.

Table 4. Comparisons of the state-of-the-art referring segmentation cIoU on RefCOCO(+/g) dataset.

Method	refCOCO			refCOCO+			refCOCOg	
	val	testA	testB	val	testA	testB	val(U)	test(U)
MCN [26]	62.4	64.2	59.7	50.6	55.0	44.7	49.2	49.4
VLT [6]	67.5	70.5	65.2	56.3	61.0	50.1	55.0	57.7
CRIS [40]	70.5	73.2	66.1	62.3	68.1	53.7	59.9	60.4
LAVT [45]	72.7	75.8	68.8	62.1	68.4	55.1	61.2	62.1
X-Decoder [50]	-	-	-	-	-	-	64.6	-
ReLA [19]	73.8	76.5	70.2	66.0	71.0	57.7	65.0	66.0
SEEM [51]	-	-	-	-	-	-	65.7	-
SESAME [42]	74.7	-	-	64.9	-	-	66.1	-
LISA-7B [15]	74.9	79.1	72.3	65.1	70.8	58.1	67.9	70.6
READ	**78.1**	**80.2**	**73.2**	**68.4**	**73.7**	**60.4**	**70.1**	**71.4**

5.5. Qualitative Results

In Fig. 3, READ qualitatively outperforms prior works, such as LISA [15] and SESAME [42] in reasoning segmentation. Remarkably, READ is capable of handling fine-grained visual grounding tasks, as illustrated in the 2^{nd} row.

5.6. Ablation Study

In this section, we conduct an ablation study to analyze the contribution of each component. We report the performance of gIoU and cIoU of *val* split on ReasonSeg dataset.

Effect of similarity as point. To evaluate the contribution of each component within the SasP module, we conduct ablation experiments focusing on three key components: the $\texttt{<SEG>}_{\text{prompt}}$, $\mathcal{P}_{\text{prompt}}$ and $\mathcal{P}_{\text{DtoC}}$. In Table 6, $\mathcal{P}_{\text{prompt}}$ significantly boosts the performace of using $\texttt{<SEG>}_{\text{prompt}}$ only. Specifically, the gIoU score increases from 51.2% to 56.4% when both prompts are used, and the cIoU score increases from 57.6% to 64.6% which validates that each component contributes crucially to the overall performance, with the combined usage leading to the best results.

Table 5. Comparisons of the state-of-the-art "see" and "segment" results on augmented FP-refcoco(+/g) *val* split. Numbers of LISA, Cascading, and SESAME are cited from [42]. "FP" is the abbreviation for False Premise, which denotes a query for an object that is absent from the provided image.

Method	FP-RefCOCO		FP-RefCOCO+		FP-RefCOCOg	
	See	Segment	See	Segment	See	Segment
LISA-7B [15]	51.36	44.00	51.32	39.62	51.25	39.64
Cascading [42]	75.59	55.18	75.03	48.64	76.07	49.98
SESAME [42]	79.84	57.93	80.00	50.81	81.78	53.79
READ	**82.87**	**61.50**	**83.51**	**54.54**	**84.67**	**56.12**

Table 6. Ablation study on similarity as points (SasP).

Exp. ID	$\texttt{<SEG>}_{\text{prompt}}$	$\mathcal{P}_{\text{prompt}}$	$\mathcal{P}_{\text{DtoC}}$	gIoU	cIoU
1		✓		37.4	29.2
2	✓			51.2	57.6
3	✓	✓		56.4	64.6
4	✓	✓	✓	**59.8**	**67.6**

6. Conclusion

In this paper, we first investigate how <SEG> token works, and we have found that what <SEG> token contributes to is the semantic similarity, akin to those of direct mentions in texts and aligns these textual semantics with its visual space upon our findings. We therefore present READ, to rectify LMMs where to "attend" when reasoning interactively by regarding similarity as points, which could be seamlessly applied to existing <SEG>-like paradigms with negligible efforts and boosts performance remarkably. For future work, we aim to actively bridge the text vocabulary embedding with its visual space by leveraging the above findings.

References

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. NeurIPS, 2022.

[2] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. In TPAMI, 2017.

[3] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Cocostuff: Thing and stuff classes in context. In CVPR, 2018.

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

[5] Xianjie Chen, Roozbeh Mottaghi, Xiaobai Liu, Sanja Fidler, Raquel Urtasun, and Alan Yuille. Detect what you can: Detecting and representing objects using holistic models and body parts. In CVPR, 2014.

[6] Henghui Ding, Chang Liu, Suchen Wang, and Xudong Jiang. Vision-language transformer and query generation for referring segmentation. In ICCV, 2021.

[7] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In CVPR, 2017.

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

[9] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In ICLR, 2022.

[10] Ronghang Hu and et al. Rohrbach, Marcus. Segmentation from natural language expressions. In ECCV, 2016.

[11] Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. Referitgame: Referring to objects in photographs of natural scenes. In EMNLP, 2014.

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

[13] Alexander Kirillov, Kaiming He, Ross Girshick, and et al. Rother, Carsten. Panoptic segmentation. In CVPR, 2019.

[14] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In ICCV, 2023.

[15] Xin Lai, Zhuotao Tian, Yukang Chen, Yanwei Li, Yuhui Yuan, Shu Liu, and Jiaya Jia. Lisa: Reasoning segmentation via large language model. In CVPR, 2014.

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

[17] 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.

[18] Feng Liang, Bichen Wu, Xiaoliang Dai, Kunpeng Li, Yinan Zhao, Hang Zhang, Peizhao Zhang, Peter Vajda, and Diana Marculescu. Open-vocabulary semantic segmentation with mask-adapted clip. In CVPR, 2023.

[19] Chang Liu, Henghui Ding, and Xudong Jiang. Gres: Generalized referring expression segmentation. In CVPR, 2023.

[20] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In CVPR, 2024.

[21] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. NeurIPS, 2024.

[22] Shu Liu, Lu Qi, and et al. Qin, Haifang. Path aggregation network for instance segmentation. In CVPR, 2018.

[23] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, and et al Su, Hang. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. arXiv preprint, 2023.

[24] Jonathan Long and et al. Shelhamer, Evan. Fully convolutional networks for semantic segmentation. In CVPR, 2015.

[25] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. arXiv:1711.05101, 2017.

[26] Gen Luo, Yiyi Zhou, Xiaoshuai Sun, Liujuan Cao, Chenglin Wu, Cheng Deng, and Rongrong Ji. Multi-task collaborative network for joint referring expression comprehension and segmentation. In CVPR, 2020.

[27] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, and et al. Yuille, Alan L. Generation and comprehension of unambiguous object descriptions. In CVPR, 2016.

[28] 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:2306.14824, 2023.

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

[30] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In ICML, 2021.

[31] Vignesh Ramanathan, Anmol Kalia, Vladan Petrovic, Yi Wen, Baixue Zheng, Baishan Guo, Rui Wang, Aaron Marquez, Rama Kovvuri, Abhishek Kadian, et al. Paco: Parts and attributes of common objects. In CVPR, 2023.

[32] Hanoona Rasheed, Muhammad Maaz, Sahal Shaji, Abdelrahman Shaker, Salman Khan, Hisham Cholakkal, Rao M Anwer, Eric Xing, Ming-Hsuan Yang, and Fahad S Khan. Glamm: Pixel grounding large multimodal model. In CVPR, 2024.

[33] Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In SIGKDD, 2020.

[34] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In TPAMI, 2016.

[35] Zhongwei Ren, Zhicheng Huang, Yunchao Wei, Yao Zhao, Dongmei Fu, Jiashi Feng, and Xiaojie Jin. Pixellm: Pixel reasoning with large multimodal model. In CVPR, 2024.

[36] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In MICCAI, 2015.

[37] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. In arXiv preprint arXiv:2302.13971, 2023.

[38] A Vaswani. Attention is all you need. In NeurIPS, 2017.

[39] Wenhai Wang, Zhe Chen, Xiaokang Chen, Jiannan Wu, Xizhou Zhu, Gang Zeng, Ping Luo, Tong Lu, Jie Zhou, Yu Qiao, et al. Visionllm: large language model is also an open-ended decoder for vision-centric tasks. In NeurIPS, 2023.

[40] Zhaoqing Wang, Yu Lu, Qiang Li, Xunqiang Tao, Yandong Guo, Mingming Gong, and Tongliang Liu. Cris: Clip-driven referring image segmentation. In CVPR, 2022.

[41] Jianzong Wu, Xiangtai Li, Xia Li, Henghui Ding, Yunhai Tong, and Dacheng Tao. Towards robust referring image segmentation. In TIP, 2024.

[42] Tsung-Han Wu, Giscard Biamby, David Chan, Lisa Dunlap, Ritwik Gupta, Xudong Wang, Joseph E Gonzalez, and Trevor Darrell. See say and segment: Teaching lmms to overcome false premises. In CVPR, 2024.

[43] Zhuofan Xia, Dongchen Han, Yizeng Han, Xuran Pan, Shiji Song, and Gao Huang. Gsva: Generalized segmentation via multimodal large language models. In CVPR, 2024.

[44] Yuwen Xiong, Renjie Liao, Hengshuang Zhao, Rui Hu, Min Bai, Ersin Yumer, and Raquel Urtasun. Upsnet: A unified panoptic segmentation network. In CVPR, 2019.

[45] Zhao Yang, Jiaqi Wang, Yansong Tang, Kai Chen, and et al. Zhao, Hengshuang. Lavt: Language-aware vision transformer for referring image segmentation. In CVPR, 2022.

[46] Qinghao Ye, Haiyang Xu, Guohai Xu, Jiabo Ye, Ming Yan, Yiyang Zhou, Junyang Wang, Anwen Hu, and et al. Shi, Pengcheng. mplug-owl: Modularization empowers large language models with multimodality. arXiv:2304.14178, 2023.

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

[48] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In CVPR, 2017.

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

[50] Xueyan Zou, Zi-Yi Dou, Jianwei Yang, Zhe Gan, Linjie Li, Chunyuan Li, Xiyang Dai, Harkirat Behl, Jianfeng Wang, Lu Yuan, et al. Generalized decoding for pixel, image, and language. In CVPR, 2023.

[51] Xueyan Zou, Jianwei Yang, Hao Zhang, Feng Li, Linjie Li, Jianfeng Wang, Lijuan Wang, and et al. Gao, Jianfeng. Segment everything everywhere all at once. In NeurIPS, 2024.

Reasoning to Attend: Try to Understand How <SEG> Token Works

Supplementary Material

We provide supplementary material related to the main paper, arranged as follows:

1. Additional implementation details (Appendix A)
2. Additional Analysis (Appendix B)
3. Additional ablation study (Appendix C)
4. Additional qualitative results (Appendix D)

A. Additional Implementation Details

A.1. Grid Search for Optimal Threshold

Given that the threshold for the foreground mask has a significant impact on the IoU, to eliminate the bias introduced by manually setting the threshold (*e.g.*, 0.5), we perform a grid search over the similarity map for each image with a step size of 0.01 to identify the optimal foreground mask. For each threshold t, we convert the similarity map into a binary mask by applying

$$\hat{\mathrm{M}}(x,y) = \begin{cases} 1 \;\; if \;\; \mathcal{S}(x,y) \geq t \\ 0 \;\; if \;\; \mathcal{S}(x,y) < t \end{cases}, \tag{14}$$

where $\mathcal{S}(x,y)$ is the similarity score for each pixel at position (x,y), $\hat{\mathrm{M}}(x,y)$ is the binary mask at that pixel. We calculate cIoU for all threshold values in the grid, and choose the threshold t' that maximizes the cIoU for the image

$$t' = \arg\max_{t} \left(\mathrm{cIoU}\left(t\right)\right). \tag{15}$$

Once the optimal threshold is selected for each image, we use it to generate the final binary masks for evaluation, which ensures that the comparison is fair and threshold-invariant.

A.2. Model Architecture and Training

As for Reasoning segmentation, we trained two models: READ-7B and READ-13B. For READ-7B, we initialize the parameters using the released SESAME model [42] to accelerate training, with the training dataset allocated in a 10:1:1:1:1:10 ratio. We employ LoRA [8] for efficient fine-tuning, using $lora\_r = 8$, and conduct end-to-end joint training. For READ-13B, we train it from scratch, using LLaVA 1.5-13B as the base model. Initially, we train it on the full dataset in a 10:10:2:3:1:1 ratio for about 8 epochs, and then fine-tune it with a ratio of 3:10:2:3:1:10, using a learning rate of 0.0001 and $lora\_r = 64$. As for referring segmentation, we maintain the same settings as those used for READ-7B in Reasoning segmentation. All our code will be publicly available at https://github.com/rui-qian/READ.

B. Additional Analysis

(1) Fig. 4 shows qualitative analysis of <SEG> token on ReasonSeg *val* split. Points derived from (a) serve as prompt with original SAM in (c). Similarity between <SEG> token and image token embeddings stemming from the last hidden layer are computed by Eq.(5), w.r.t. LLaVA encoder in (a) and SAM decoder in (b). The consistency in (a), (b) indicates that <SEG> token in LMMs learns semantics similar to those of direct mention in texts, as observed in CLIP [30]. Note that 1^{st} column in (b) shows failure cases, indicating the existence of misalignment between the LLaVA encoder in (a) and SAM decoder in (b). Such observation sheds light on the interpretability of semantic alignment issues, where the LLaVA encoder generates accurate textual responses even in scenarios where the SAM decoder fails at segmentation, when eliciting LISA [15] for reasoning explanations. In future work, we aim to further investigate the underlying connections behind this phenomenon. (2) Fig. 5 shows qualitative analysis of $\mathcal{P}_{\text{prompt}}$ on ReasonSeg *val* split. We first select several points with the highest similarity scores as positives (red in (b)) and an equal number of points with the lowest similarity scores as negatives (blue in (b)). These points are then directly used as prompts, rather than the <SEG> token, and input into the original SAM model to generate the segmentation mask. Columns in (b) demonstrate that only relying on the selected similarity points as prompt can still generate a segmentation mask potentially.

C. Additional Ablation Study

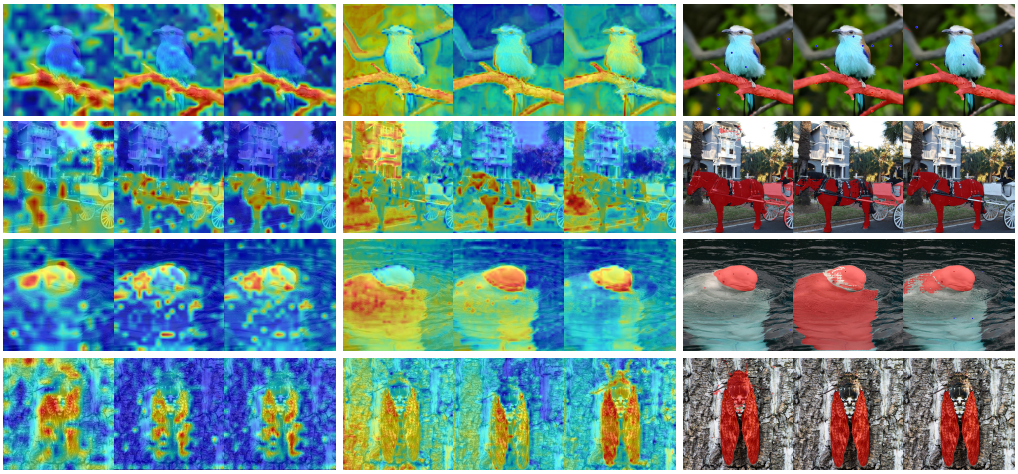
Effect of points ratios. To explore how the ratios of positive, negative, and neutral points impact the performance of READ, we vary the positive and negative thresholds (t_{pos} and t_{neg}) as well as the number of points $|\mathcal{P}|$. As the positive sample ratio (t_{pos}) increases, model performance improves, particularly when fewer points are used ($|\mathcal{P}|$= 10). Also, increasing the number of points generally enhances performance, with the most significant improvements observed at $|\mathcal{P}|$= 60, regardless of the t_{pos} setting.

Table 7. Ablation study on points ratios.

		$\lvert\mathcal{P}\rvert$=10		$\lvert\mathcal{P}\rvert$=30		$\lvert\mathcal{P}\rvert$=60	
t_{pos}	t_{neg}	gIoU	cIoU	gIoU	cIoU	gIoU	cIoU
0.8	0.2	58.94	65.16	**59.75**	67.62	59.71	**68.17**
0.7	0.3	58.48	64.00	58.82	65.32	59.20	67.70
0.6	0.4	58.59	64.27	58.66	65.00	58.93	66.69

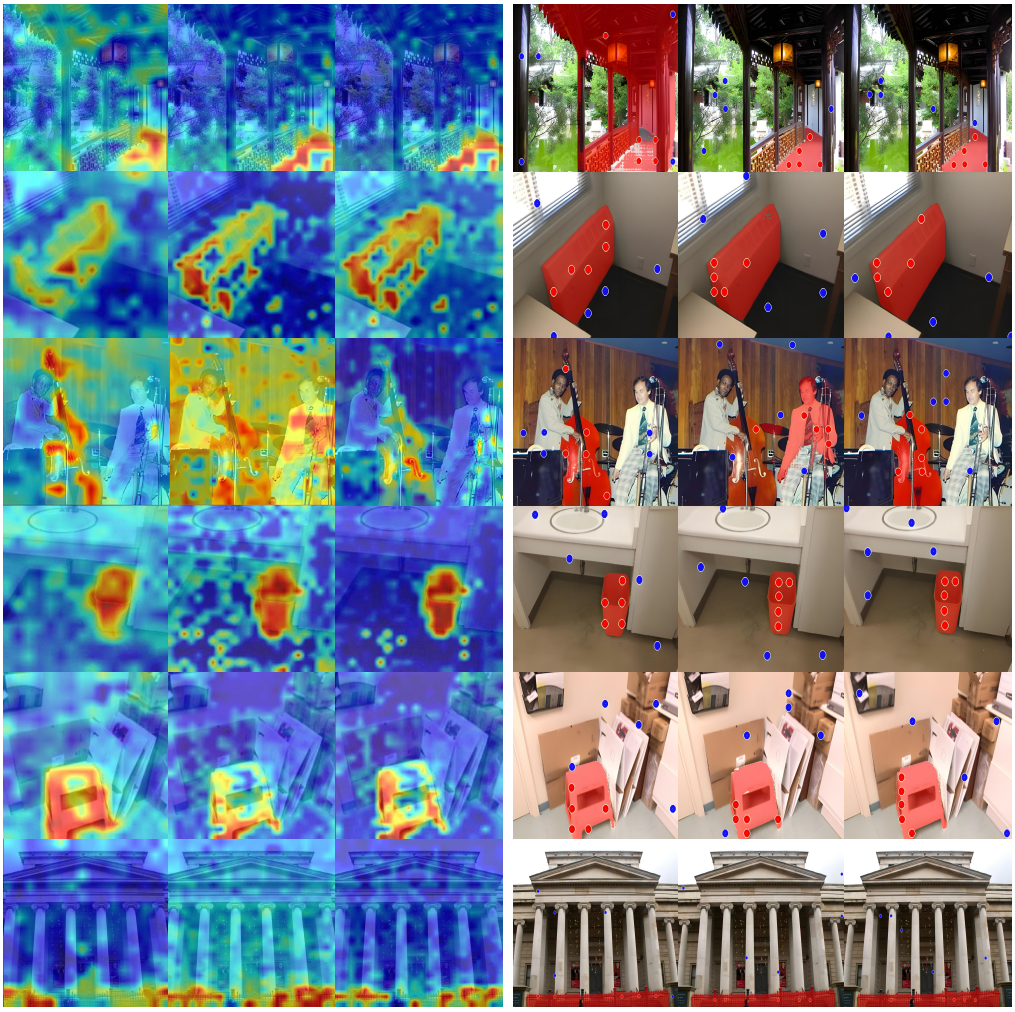
D. Additional Qualitative Results

Fig. 7 shows the qualitative results of READ on ReasonSeg *val* split. LISA and SESAME exhibit various defects to some extent when handling the displayed cases, whereas our approach delivers more desirable segmentation results. Fig. 6 show qualitative results on FP-RefCOCO(+/g) *val* split. Also, READ retains LLMs' conversational ability while performing segmentation tasks and can refuse to output a mask when the queried object doesn't exist.



(*a*) <SEG> with LLaVA (*b*) <SEG> with SAM (*c*) Points as prompt with SAM

Figure 4. Qualitative analysis of <SEG> token on ReasonSeg *val* split. The 1^{st}, 2^{nd}, and 3^{rd} columns of (a), (b), and (c) are LISA, SESAME, and READ(Ours) for comparisons, respectively.



(*a*) <SEG> with LLaVA (*b*) Points as prompt with SAM

Figure 5. Qualitative analysis of $\mathcal{P}_{\text{prompt}}$ (points as prompt) on ReasonSeg *val* split. The 1^{st}, 2^{nd}, and 3^{rd} columns of (a), (b) are LISA, SESAME, and READ(Ours) for comparisons, respectively.

In this image, is **right sheep** visible? If so, produce the segmentation map. If not, reject and list any items that may look like right sheep.

Right sheep is spotted. Segmentation map: [SEG] .

Confirm the presence of **left sheep** in the image. If it exists, show the segmentation. Otherwise, deny and, as applicable, specify objects that resemble left sheep.

Left sheep is spotted. Segmentation map: [SEG] .

Is **baby goat** detectable in this image? If yes, show the segmentation map. If no, explicitly deny and optionally highlight similar items.

Baby goat is spotted. Segmentation map: [SEG] .

Please check for **dragon** on the left upper corner in the image. Share a segmentation map if it exists. Otherwise, confirm its absence and, where applicable, identify objects resembling dragon on the left upper corner.

No visual output

I **don't find dragon** on the left upper corner, but I do see sheep on the left upper corner.

Examine this image for the presence of **baseball player** in red shirt to right of batter. If detected, supply a segmentation map. If not, deny its presence.

Yes, **baseball player** in red shirt to right of batter is right here. Segmentation: [SEG] .

Is there an occurrence of **20 guy** in this image? If yes, display the segmentation map. If no, reject and identify any resembling objects.

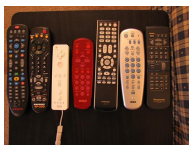
20 guy is definitely in the image. Segmentation: [SEG] .

Could you check this image for the existence of **soccer player** in red shirt to right of batter? If found, show the segmentation map. If not found, reject and, if suitable, point out similar objects.

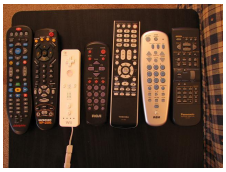
No visual output

The **soccer player** in red shirt to right of batter **isn't part of this**, but there is baseball player in red shirt to right of batter.

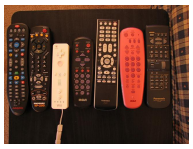
Please verify the existence of **fourth remote from left** in this image. Present a segmentation map if applicable. Otherwise, refute and, where relevant, indicate any potentially confusing objects.



Yes, I see it. Segmentation map for **fourth remote from left**: [SEG] .



Please analyze this image for any trace of **second remote from right**. If detected, provide the segmentation map. If not, deny and, if relevant, mention any objects that could be confused with second remote from right.



Yes, **second remote from right** is right here. Segmentation: [SEG] .

Is there **a banana on rightside** visible in this image? If yes, I'd like to see the segmentation map. If not, please deny its presence and, as needed, note any similar items present.

No visual output

I **don't see banana** on rightside, but remote on rightside is present.

Figure 6. Visualization on FP-RefCOCO(+/g) *val* split.

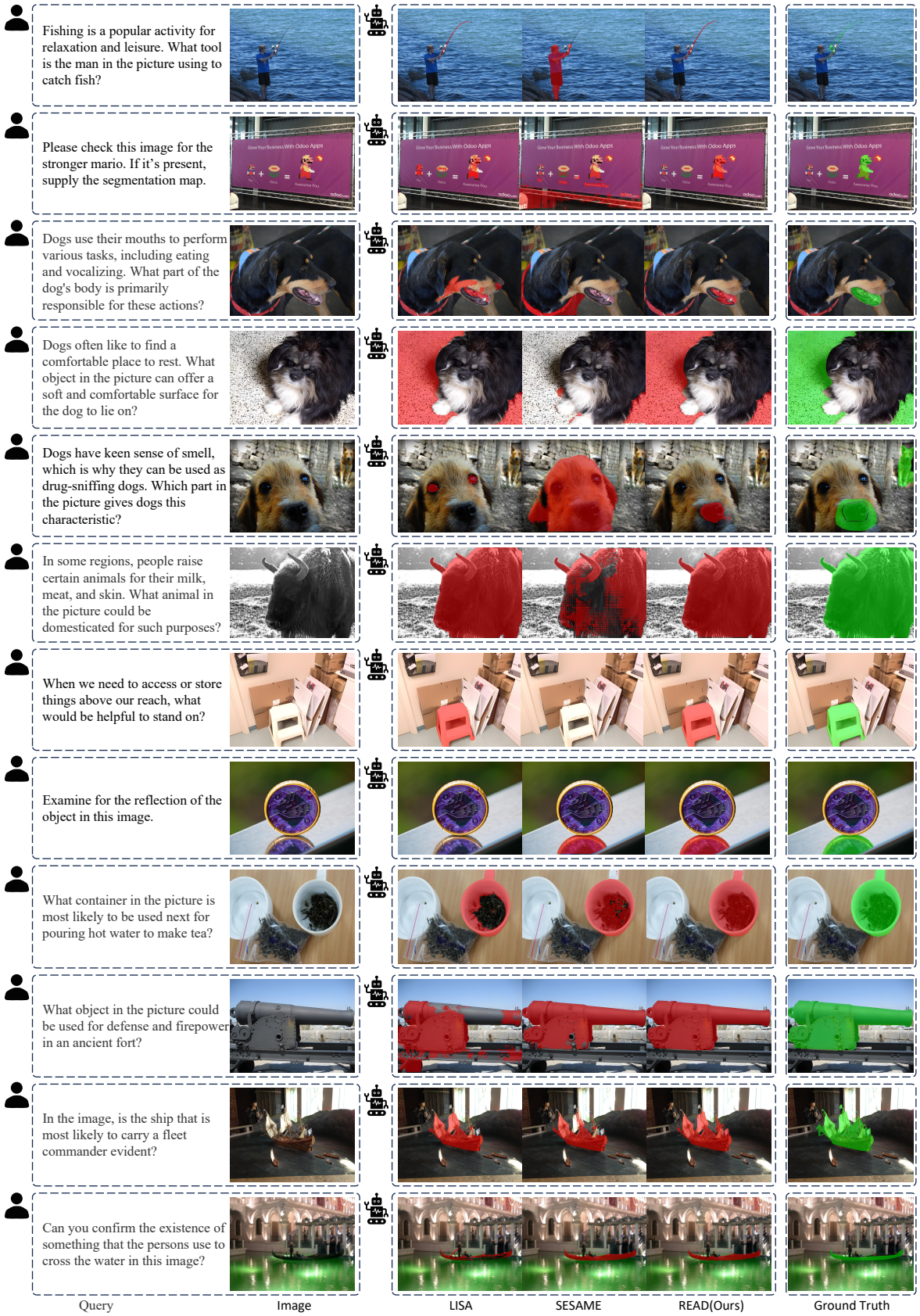


Figure 7. Visual comparison among READ (ours) and prior works on ReasonSeg *val* split.