PSALM: Pixelwise SegmentAtion with Large Multi-Modal Model

Zheng Zhang[1,2*], Yeyao Ma[1*], Enming Zhang[1*], and Xiang Bai[1✉]

[1] Huazhong University of Science and Technology
[2] Microsoft Research Asia

Abstract. PSALM is a powerful extension of the Large Multi-modal Model (LMM) to address the segmentation task challenges. To overcome the limitation of the LMM being limited to textual output, PSALM incorporates a mask decoder and a well-designed input schema to handle a variety of segmentation tasks. This schema includes images, task instructions, conditional prompts, and mask tokens, which enable the model to generate and classify segmentation masks effectively. The flexible design of PSALM supports joint training across multiple datasets and tasks, leading to improved performance and task generalization. PSALM achieves superior results on several benchmarks, such as RefCOCO/RefCOCO+/RefCOCOg, COCO Panoptic Segmentation, and COCO-Interactive, and further exhibits zero-shot capabilities on unseen tasks, such as open-vocabulary segmentation, generalized referring expression segmentation and video object segmentation, making a significant step towards a GPT moment in computer vision. Through extensive experiments, PSALM demonstrates its potential to transform the domain of image segmentation, leveraging the robust visual understanding capabilities of LMMs as seen in natural language processing. Code and models are available at `https://github.com/zamling/PSALM`.

Keywords: Segmentation · Large Multimodal Model · Visual Language

1 Introduction

Large multi-modal model (LMM) has ignited the dawn of the vision GPT [2] moment by making ground-breaking progress in various advanced visual understanding tasks by compressing image and language information into a single auto-regressive model. However, there are still many obstacles on the road to achieving vision GPT, and an important one being that current LMM can only perform text outputs, making it challenging to address the pixel-level image understanding problem directly, *i.e.*, image segmentation, which is one of the most critical tasks in computer vision.

Behind the obstacle is many challenges. First, the default output of the LMM is discrete tokens, and there is no apparent way to generate masks directly.

* Equal contribution. ✉ Corresponding author.

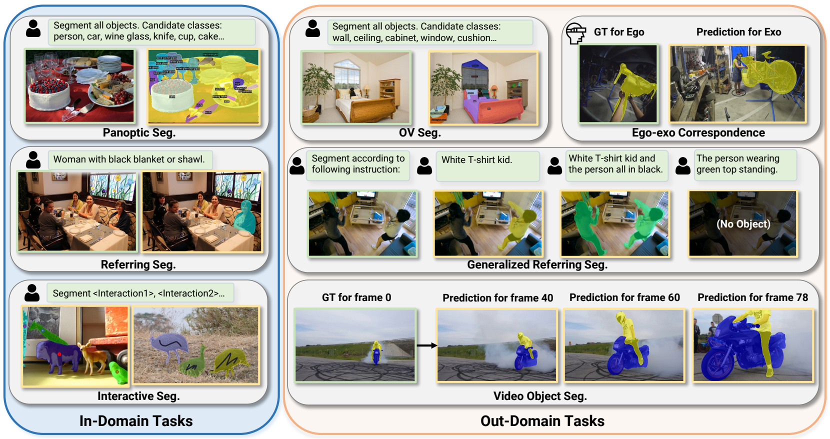

Fig. 1: PSALM has capability to handle multiple segmentation tasks in only one single model. We visualize some tasks, including Panoptic segmentation in COCO [24]; Open-Vocabulary instance segmentation in ADE20K [59]; Interactive segmentation in COCO-Interactive; Referring segmentation in RefCOCO [57]; Generalized referring segmentation in gRefCOCO [25]; Ego-exo correspondence in Ego-Exo4d [12]; Video object segmentation in DAVIS2017 [36].

Second, the variety of image segmentation tasks requires different forms of inputs and outputs. For example, semantic segmentation needs support inputs with different categories. Instance segmentation demands generate object IDs and the class confidence score for each object. Referring segmentation is supposed to have a language sentence as input and interactive segmentation has more varied inputs than the other tasks, which can be points, scribbles, bounding boxes, or masks. Third, unifying different segmentation tasks with a shared weight model is also challenging because different tasks require varied capabilities.

In this work, we propose a method named PSALM (**P**ixelwise **S**egment**A**tion with **L**arge Multi-Modal **M**odel) that aims to address the above challenges and extend the capabilities of LMM from text-output tasks to general segmentation tasks (Fig. 1 shows representative tasks). Specifically, PSALM externalizes a mask decoder on the top of LMM and designs a flexible input schema to unify different segmentation tasks into a single model.

The input schema consists of four different parts: images, task instruction prompt, condition prompt, and a set of mask tokens, where the instruction prompt is a text sentence describing the task itself, condition prompt contains the additional necessary information to solve the task, either in terms of category names, sentence or visual features, and mask tokens are a set of learnable embeddings. All these inputs are fed into the LMM, and the resulting output mask tokens are further used as input by the mask generator to present mask proposals. Apart from producing the mask proposals, it is also necessary to predict the class of each segmentation mask or estimate a confidence score, which can be achieved by using the output embedding of the condition prompt as the classifier weights to classify each mask proposal. In practice, we categorize conditions into category condition, sentence condition, and visual-prior condition, and present the corresponding methods to build the classifier weights according to the properties of each type of condition.

Some other methods, represented by LISA [17], also aim to use LMM for segmentation tasks. However, these methods are usually designed for referring segmentation and fail to justify their ability to solve generalized segmentation tasks (see Tab. 1). In contrast, thanks to the generality and flexibility of the proposed architecture, PSALM can not only solve a variety of segmentation tasks but also be able to joint train on different tasks, which makes the model task generalizable while allowing the model to take full advantage of the intrinsic connections of different datasets/tasks to achieve better performance. Specifically, with the joint training of COCO Panoptic Segmentation [24], RefCOCO [57]/RefCOCO+/RefCOCOg [31], and COCO Interactive, we observe a significant performance improvement compared to training at different tasks individually, and therefore result in even better performance than other task-specific methods. On referring segmentation tasks, we outperform other LLM-based pixel reasoning methods (*e.g.*, LISA, PixelLM [40] and GSVA [48]) on RefCOCO, RefCOCO+, and RefCOCOg, and it worth noting that we only use Phi-1.5 1.3B model [22] while others adopt Vicuna-7B [7] or LLama2-13B model [45].

The flexible design of architecture and input schema, multi-task joint-training, and the strong visual understanding capability of LMM not only make PSALM perform well on trained in-domain tasks but also enable generalizability to out-of-domain tasks in a zero-shot manner, *i.e.,* directly dealing with unseen tasks without additional training. We test on three tasks: open-vocabulary segmentation, generalized referring expression segmentation, and video object segmentation. PSALM achieves promising zero-shot performance on these tasks. We believe this task-level generalizability is crucial, which is one of the key properties of the large language model for its success in natural language processing.

Through extensive experiments on a variety of segmentation tasks, we show that presented PSALM has strong potential to address general image segmentation tasks and exhibits a certain degree of task generalization capability as LLM does in NLP. We believe that this work can inspire extensive research on realizing the GPT moment in computer vision and facilitate its arrival.

2 Related works

2.1 Large Multimodal Models

With the release of GPT-V [32] and Gemini [44], more attention and efforts from open-source and research communities are shifting from large language models (LLM) to large multi-modal models (LMM). LLaVA [27], BLIP-2 [19], and Flamingo [1] are three representative works, where the core idea of both LLaVA and BLIP-2 is to map visual features into the input space of LLM to implement multi-modal capabilities, while Flamingo employs deeper feature fusion in the intermediate layers of LLM. Some works, such as Kosmos-2 [33],

Table 1: Capability for different methods. Our proposed PSALM can handle more segmentation tasks than other LLM-centric methods. LLM-centric methods can also deal with text generation tasks, which is hard for most vision-centric methods.

	Methods	Generic Seg.	Referring Seg.	Interactive Seg.	OV Seg.
Vision centric	Mask2Former [6]	✓			
	ODISE [49]	✓			✓
	UNINEXT [53]	✓	✓	✓	
	SEEM [61]	✓	✓	✓	✓
	OMG-Seg [21]	✓		✓	✓
LLM centric	LISA [17]		✓		
	GLAMM [39]		✓		
	PixelLM [40]		✓		
	PSALM (Ours)	✓	✓	✓	✓

Shikra [4], and Ferret [55], further introduce object localization tasks into the LMM, while others, such as Emu [43], CogVLM [46], and DreamLLM [10], focus on how to integrate visual generation into the LMM. Monkey [23], OtterHD [18], and LLaVA-NeXT [26] explores mechanisms for processing large-size images and significantly improves performance on tasks such as OCR. However, these above methods are mainly designed for text output tasks or image generation and cannot directly deal with pixel-level understanding tasks such as image segmentation, which is different from ours and we can base on these models.

2.2 Pixel Reasoning with LMM

Similar to our goal, some existing works attempt to enable LMMs to generate segmentation masks. LISA [17] is a pioneering work that uses a special seg token to aggregate information of a given sentence and use it as a prompt embedding in a SAM decoder to predict the segmentation mask. u-LLaVA [50] further supports object grounding tasks on the basis of LISA, and NExT-Chat [58] introduces richer inputs, such as bounding boxes. Furthermore, since LISA can only deal with a single object, many subsequent works that attempt to address the multi-object case, such as GLaMM [39], PerceptionGPT [34], PixelLM [40], GSVA [48], and LISA++ [54], all of which share the basic idea of introducing a seg token for each sentence describing a different object, and except PixelLM, all other methods are based on SAM decoder [15].

Although all these methods can generate masks, they are primarily designed for reference segmentation. In contrast, our method is designed for generalized segmentation tasks, which have diverse input and output requirements. In addition, the difference in goals also brings technical discrepancies: these methods use language models to directly generate the final segmentation masks, while our approach is closer to Mask2Former [6], which first generates mask proposals and then classifies the masks.

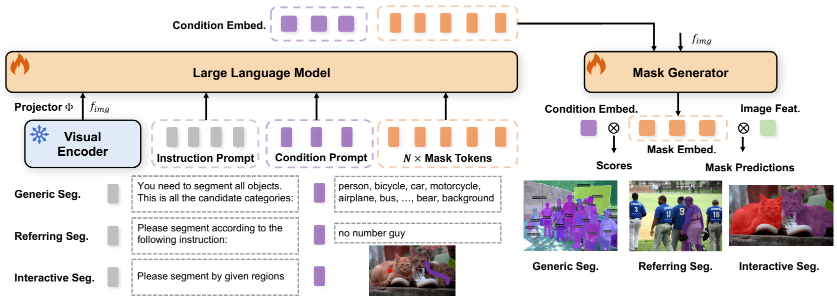

Fig. 2: PSALM architecture overview.

2.3 Unified Segmentation Framework

Another class of relevant explorations [21, 47, 53, 61] studies how to unify different segmentation tasks into a single framework. For instance, Mask2Former [6] presents a unified architecture of generic segmentation [3] but with different models. OneFormer [14] further integrates these three tasks within a single model. UNINEXT [53] aims to unify instance-perception tasks and introduce text inputs and thus it can handle referring segmentation. X-Decoder [60] presents a flexible decoder architecture that can support generic segmentation, referring segmentation, retrieval, and image captioning. SEEM [61] designs a generic encoder-decoder to unify different segmentation tasks, where the encoder is used for projecting image, text, and human inputs into a joint visual-semantic space, and the decoder is used for mask prediction. However, all these works are not based on LMM, instead, they are mostly vision-centric models, *i.e.*, usually designed for visual tasks only and thus cannot address language tasks very well.

3 Methods

Fig. 2 provides an overview schematic of PSALM, which consists of a large multimodal model (LMM), a mask generator, and a flexible input schema designed for general segmentation tasks. The input schema has four different types of inputs: image, task instruction prompt, condition prompt, and a set of mask tokens. LMM processes the input tokens and the output embedding of mask tokens is further fed into the mask generator to generate masks. In the following, we will introduce our approach in detail.

3.1 Large Multimodal Model and Input Schema

PSALM is built on large multimodal models (LMM), and there are many different LMM architectures, such as LLaVA [27], BLIP [20], and Flamingo [1]. Here,

[3] Generic segmentation includes semantic segmentation, instance segmentation, and panoptic segmentation.

we adopt the design of LLaVA because of its proven performance and simplicity, but the other LMM architectures are also compatible with our approach without any theoretical difficulties.

The LMM used in our work has a visual encoder and pre-trained large language model (LLM). The two models are connected by a lightweight vision-language alignment model, which is a 3×3 convolution layer followed by a linear layer. The official LLaVA model uses a frozen CLIP model [37] as a visual coder, whose features lack the fine-grained information that is required for segmentation tasks [52]. Therefore, we train a customized LLaVA model by using the Swin Transformer [28], and limited by resources, we additionally replace the LLM from the Vicuna 7B model [7] to a smaller Phi-1.5 1.3B model [22]. Here, we applied only the first visual-language alignment stage of LLaVA by following its default settings. In our ablations, we found the alignment stage is crucial for open-vocabulary segmentation and referring segmentation tasks (Tab. 6).

Different segmentation tasks need different forms of inputs and outputs, which motivates us to present a flexible input schema to unify various requirements. In addition to the input image used in the visual encoder, our input schema has three other different types of inputs: task instruction prompt, condition prompt, and a set of mask tokens. We will introduce each of them and summarize the prompts used for all different tasks in the Appendix.

Task Instruction Prompt. The task instruction prompt is usually a text sentence describing and specifying the model's task. For example, in panoptic segmentation, the task instruction can be *"You need to segment all objects. This is all the candidate categories."* and in referring segmentation, the instruction can be *"Please segment according to the following instruction."*

Condition Prompt. Some tasks require additional information to perform, *e.g.*, panoptic segmentation needs specifying the candidate set of categories to be segmented, and interactive segmentation needs interactive inputs. The condition prompt is designed for these tasks. In addition to providing information, condition prompt also plays an important role in predicting categories or estimating confidence scores for each segmentation mask. In Sec. 3.3, we will discuss the design of condition prompts for different tasks in detail.

Mask Token. The LLM is designed for text output and cannot directly generate segmentation masks. To bypass this challenge, we append a set of mask tokens after other inputs, and then these mask tokens are decoded to segmentation masks by a mask generator (will be introduced in Sec. 3.2). This design is inspired by Mask2Former [6], with the difference that the mask tokens in Mask2Former are used directly in the mask generator, whereas the mask tokens in our approach are first updated by the LMM and then used in the mask generator, and we found our approach leads better performance in practice (see Tab. 2).

Some works, such as LISA [17] and PixelLM [40], take similar seg tokens as input and employ a decoder to generate the masks. However, our objective is fundamentally different: in LISA and PixelLLM, seg tokens are used to generate the final prediction, while we generate mask proposals and further classify them based on condition prompts. Compared to the design of LISA and PixelLLM,

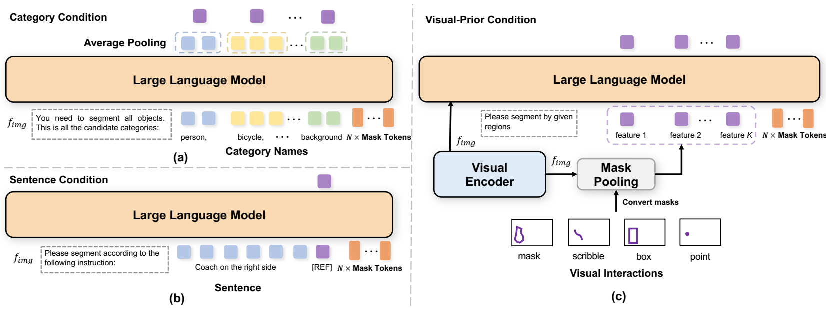

Fig. 3: Detailed processing for different condition prompts. (a) shows the processing for category condition. (b) shows the processing for sentence condition. (c) shows the processing for visual-prior condition.

our approach is more flexible and adaptable to more tasks, yet decoupling mask generation and classification is more efficient (see Tab. 3 and Tab. 4).

3.2 Mask Generator

The mask generator predicts the mask and their category probabilities from three inputs: a multi-level visual features $\{v_l\}_{l=1}^{L}$, a set of mask tokens $\{q_i\}_{i=1}^{N}$, and a set of condition embeddings $\{c_k\}_{k=1}^{K}$. It can be formally defined as:

$$\{(m_i, p_i)\}_{i=1}^{N} = \texttt{MaskGenerator}(\{v_l\}_{l=1}^{L}, \{q_i\}_{i=1}^{N}, \{c_k\}_{k=1}^{K}), \tag{1}$$

where $m_i \in \mathbb{R}^{H\times W}$ is the i-th predicted segmentation mask and $p_i \in \mathbb{R}^{K}$ is the corresponding category probability. In practice, the multi-level visual features are the internal features of the Swin visual encoder used in LMM. The design of the mask generator follows Mask2Former, which employs multi-scale deformable attention as a pixel decoder and a transformer-based mask decoder to generate segmentation masks. The class of each mask is predicted by the condition embedding $\{c_k\}$, which is basically obtained from the output of the condition prompt, the obtaining method is slightly different for different types of conditions.

3.3 Design of Condition Prompts

In our approach, the conditional prompt serves two important purposes: First, it provides the necessary information required to solve the task; Second, we use the output embedding of the conditional prompt in the LLM as classifier weights to predict the class of each segmentation mask. The design of the conditional prompt is closely tied to the type of task, and based on the information required for different tasks, we summarize three condition types: category condition, sentence condition, and visual-prior condition.

Category Condition This condition type is used for tasks that need specifying a set of categories to be segmented, such as semantic segmentation, instance segmentation, and panoptic segmentation, and often needs to predict the class probability of each segmentation mask. Specifically, given a set of category names, we join them into a sentence by a comma separator, *e.g.,*, given three categories: person, bicycle, and car, the joint sentence is *"person, bicycle, car"*.

The joint sentences are then proceeded by LMM to get the output embeddings, which can be further used to classify the predicted segmentation masks. Specifically, for each category, we select its corresponding output embeddings and apply the `avg_pooling` over them to obtain a condition embedding $c \in \mathbb{R}^D$, where D is the embedding dimension, and thus condition embedding for all categories is a set $\{c_k\}_{k=1}^K$, where K is the number of categories (see Fig. 3 (a)). This embedding set can be used by the mask generator to predict the class.

Sentence Condition This condition is usually used for referring segmentation. Unlike the category condition, where category names are usually short, sentences are much longer, and not every word in the sentence is useful, so the `avg_pooling` is not the optimal choice here. Instead, we introduce a special `[REF]` token, which is appended after the condition sentence as an anchor to aggregate useful information, and the output embedding of `[REF]` token, *i.e.,* the output features of LMM on the location of `[REF]` token, is used as condition embedding c and used by mask generator, as shown in Fig. 3 (b).

Visual-Prior Condition We formulate most interaction (*e.g.,* point, mask, box, or scribble) used in interactive segmentation tasks as the visual-prior condition. Taking the scribble as an example, we first generate a binary scribble curve with a width of 5 pixels on a binary map, whose size is the same as the input image of the Swin vision encoder, and then apply the binary map to the output features of the vision-language alignment model, performing average mask pooling by upsampling the output feature map to the size of the binary map. The average pooled feature is used as the visual-prior condition and fed into LMM. If there are multiple interactions, the process is repeated for each, yielding multiple pooled features as inputs, each feature separated using a strategy akin to the category condition. For other types of interaction, we adopt similar approaches. Specifically, for box or mask, we directly apply the pooling operator by treating them as binary masks, and for point, we bold it to a 10-pixel circle and then apply the mask pooling. After that, we use the output embedding of the visual-prior condition as the classifier weight in the mask generator to estimate the confidence of each mask proposal, as shown in Fig. 3 (c).

3.4 Training Objectives

The training process of PSALM has two stages: In the first stage, we train the visual language alignment model following LLaVA setting by freezing the vision encoder and LMM; In the second stage, we only freeze the vision encoder and fine-tune all other modules, including the mask generator. Similarly to Mask2Former, we use matching loss in the second stage training, *i.e.*, we use bipartite matching to find the optimal assignments between the mask proposals and the ground truth masks by minimizing the matching loss and use these assignments to perform training. The loss has two terms: $\mathcal{L} = \mathcal{L}_{mask} + \mathcal{L}_{cls}$, where $\mathcal{L}_{mask}$ indicates the mask loss which comprises a pixel-level Binary Cross-Entropy (BCE) loss and Dice loss [42], while the $\mathcal{L}_{cls}$ indicates the category classification loss, and we use Cross-Entropy loss for category condition and BCE loss for other cases.

4 Experiments

We conduct extensive experiments on various datasets and tasks to analyze PSALM. Details of the datasets and implementations are given in the Appendix.

4.1 Ablations

We first ablate key designs and present behind insights in this section. To better show how different designs affect the performance on a wide range of tasks, we mainly report the results on three in-domain benchmarks: COCO Panoptic Segmentation (COCO-Pan), RefCOCO-val (RefCOCO), and COCO Interactive Segmentation with point inputs (COCO-Point), and one out-of-domain benchmark: open-vocabulary instance segmentation on ADE20K-150 (A150-OV).

Table 2: Ablation on the design of mask tokens. w.LLM: use mask tokens as inputs of LLM. Prefix: place mask tokens at the front. Suffix: place mask tokens at the end.

Mask Tokens		COCO-Pan	RefCOCO	COCO-Point	A150-OV
w.LLM	Pos.	PQ	cIoU	mIoU	mAP
✓	Suffix	55.1	76.1	53.3	9.3
✗	-	54.8(↓0.3)	74.3(↓1.8)	53.1(↓0.2)	8.2(↓1.1)
✓	Prefix	55.0(↓0.1)	75.1(↓1.0)	53.0(↓0.3)	7.8(↓1.5)

Table 3: Effect of decouple design in COCO Semantic Segmentation.

Decouple	mIoU	fwIoU
✓	66.5	72.5
✗	42.7	35.0

Design of Mask Tokens. In our approach, we use a set of mask tokens to predict the mask proposal. In practice, we have found that using mask tokens as inputs to the LLM leads to better performance than applying them directly to the mask generator, which is the default method of Mask2Former. Tab. 2 shows the results, where the direct use of mask tokens leads to a noticeable performance degradation in RefCOCO and A150-OV. We believe this is because using mask tokens as input leads to a better awareness of the information needed for the task and thus improves performance, which is essential for these two tasks. For better validation, we placed mask tokens before conditional prompts and task instruction prompts and found a similar performance drop to that of not using mask tokens in LMM, which further supports our hypothesis.

Compared to LISA and other methods that use seg token to generate final segmentation results directly, our mask proposal approach has three advantages: First, our design is more flexible and thus can be applied to a wider range of segmentation tasks, especially tasks that require predicting category or confidence scores; Second, our design decouples the mask prediction and classification, which alleviating the learning difficulties for some tasks. In Tab. 3, we study how the decouple design affects the semantic segmentation performance on COCO Semantic Segmentation[4], and we found our decouple design is significantly better[5]. Third, the mask proposals allow multiple masks to be generated for a single instance, which makes the mask accuracy superior to solutions like LISA that only predict a single mask. Tab. 4 shows that using more mask proposals on RefCOCO gives a clear improvement over using a single mask.

Table 4: Effects of number of mask tokens in RefCOCO(cIoU). The results of LISA are listed as a reference.

#Mask Token	val	testA	testB
LISA	74.1	76.5	71.1
1	75.3	78.0	72.2
100	76.5	78.5	73.4

Table 5: Ablation on different designs for condition prompts.

Condition		COCO-Pan	RefCOCO	A150-OV
Category	Sentence	PQ	cIoU	mAP
`avg_pooling`	`avg_pooling`	55.1	75.3	9.2
`[REF]`	`[REF]`	54.9	76.1	8.4
`avg_pooling`	`[REF]`	55.1	76.1	9.3

Design of Condition Prompts. Another key design in PSALM is the condition prompt, particularly the way we obtain the condition embeddings, which are used as classifier weights in the mask generator to predict the class of mask proposals. As described in Sec. 3.3, for category condition, we use the `avg_pooling` over output embeddings of each class name as the condition embedding, and for sentence condition, we adopt a `[REF]` token to aggregate useful information. Tab. 5 shows the ablation, where we first tried to use the same design for all conditions and found that `avg_pooling` performed slightly better on COCO-Pan, with a larger improvement on A150-OV, while `[REF]` worked better on RefCOCO. We further used different designs and found that the advantages of each design are preserved, and the best overall performance is achieved.

Table 6: Ablation on effect of vision-language alingment.

VL Alignment	COCO-Pan PQ	RefCOCO cIoU	COCO-Point mIoU	A150-OV mAP
✓	55.1	76.1	53.3	9.3
✗	54.9(↓0.2)	71.7(↓4.4)	53.0 (↓0.3)	8.2(↓1.1)

Table 7: Ablation on joint training.

Ablation	COCO-Pan PQ	RefCOCO cIoU	COCO-Point mIoU	A150-OV mAP
Task Specific Train	55.6	76.5	62.2	7.0
Joint Train	55.9	83.6	64.1	9.0
Δ	+0.3	+ 7.1	+ 1.9	+2.0

[4] This benchmark is introduced by Mask2Former, which is composed by merging instances belonging to same class together in COCO Panoptic Segmentation.
[5] More experimental details are in Appendix.

Table 8: Comparison with the state-of-the-art methods on three referring image segmentation benchmarks with cIoU. (ft) denotes models further finetuned on RefCOCO/+/g after mix training. We abbreviate the datasets: COCO(C) [24], LVIS(L) [13], RefCOCO(RC) [57], Object365(O365) [41], Video segmentation datasets(V), ADE20K(A) [59], COCO-Stuff(CS) [3], PACO-LVIS(PL) [38], PASCAL-Part(PP) [5], GranD(G) [39], VOC2010(VOC) [11], Visual Genome(VG) [16], Flicker30k(F30K) [35], MUSE(M) [40], gRefCOCO(gRC) [25], COCO-Interactive(CI).

Method	Segmentation Data	LLM Type	RefCOCO			RefCOCO+			RefCOCOg	
			val	testA	testB	val	testA	testB	val	test
SEEM-L [61]	C, L, RC	-	-	-	-	-	-	-	65.6	-
UNINEXT-L [53]	O365, C, RC, V	-	80.3	82.6	77.8	70.0	74.9	62.6	73.4	73.7
UNINEXT-H [53]	O365, C, RC, V	-	82.2	83.4	81.3	72.5	76.4	66.2	74.7	**76.4**
LISA [17]	A, CS, RC, PL, PP	Vicuna-7B	74.1	76.5	71.1	62.4	67.4	56.5	66.4	68.5
LISA(ft) [17]	A, CS, RC, PL, PP	Vicuna-7B	74.9	79.1	72.3	65.1	70.8	58.1	67.9	70.6
GLaMM [39]	G, RC	Vicuna-7B	79.5	83.2	76.9	72.6	78.7	64.6	74.2	74.9
u-LLaVA [50]	A, CS, RC, PL, VOC	Vicuna-7B	80.4	82.7	77.8	72.2	**76.6**	66.8	**74.8**	75.6
PerceptionGPT [34]	RC, VG, F30k	Vicuna-13B	75.3	79.1	72.1	68.9	74.0	61.9	70.7	71.9
PixelLM [40]	A, CS, RC, PL, M	Llama2-13B	73.0	76.5	68.2	66.3	71.7	58.3	69.3	70.5
GSVA [48]	A, CS, RC, PL, PP, gRC	Llama2-13B	77.7	79.9	74.2	68.0	71.5	61.5	73.2	73.9
GSVA(ft) [48]	A, CS, RC, PL, PP, gRC	Llama2-13B	79.2	81.7	77.1	70.3	73.8	63.6	75.7	77.0
PSALM	C, RC, CI	Phi-1.5 (1.3B)	**83.6**	**84.7**	**81.6**	**72.9**	75.5	**70.1**	73.8	74.4

Importance of VL-alignments. The visual-language alignment stage (*i.e.,* first training stage) is to project the visual features to the text input space, and it is an important step towards making the LLM understand images. In Tab. 6, we examine the impact of this stage, and we found that without VL alignment, the performance of all four tasks becomes worse, with the performance of A150-OV and RefCOCO being significantly affected, for example, the A150-OV dropped by -1.1 mAP and RefCOCO even dropped by -4.4 cIoU, probably because these two tasks require a strong requirement on understanding the relationship between vision and language. This result also suggests that the VL alignment is essential, and the LMM-based segmentation models have strong potential.

Joint Training. Our architecture design and input schema help integrate various segmentation tasks so that they can be trained on one model. Tab. 7 shows the effect of this joint training on different tasks. For the task-specific models, we perform the training on the corresponding task data for 18k iterations. In contrast, the joint training setting (see implementation section for details) has a total of 56k training iterations, which corresponds to 14k iterations per task. The results show that joint training of different tasks greatly improves the performance. This suggests that learning between tasks is mutually beneficial, which is also the secret of success in LLM. For example, the generic segmentation task helps refine the mask prediction in the referring segmentation, and referring expressions also enhance the model's ability to recognize more unseen categories, which in turn improves the performance of open-vocabulary segmentation tasks.

4.2 System-Level Comparison on In-Domain Tasks

In this section, we compared our model on three in-domain tasks with other state-of-the-arts to illustrate the effectiveness of our approach.

Referring Segmentation. Most existing works aimed at getting LMMs to perform image segmentation are designed for reference segmentation tasks. We compare PSALM with other works on RefCOCO, RefCOCO+, and RefCOCOg, and Tab. 8 shows the results. Owing to the generalized and flexible design of PSALM and the advantages of joint training on multiple tasks and datasets, our system was able to achieve state-of-the-art (SOTA) performance on RefCOCO and RefCOCO+, and competitive performance on RefCOCOg with STOA, despite being driven by LLM with only 1.3B parameters. It is worth noting unlike methods such as LISA and GSVA, which may achieve improvements through task-specific fine-tuning (gray-labeled results), PSALM does not perform additional fine-tuning but still achieves better performance on RefCOCO and RefCOCO+ than their fine-tuned models.

Generic Segmentation. We evaluate PSALM with state-of-the-art methods on the COCO Panoptic Segmentation validation set (Tab. 9). Here, we follow the evaluation protocol used in Mask2Former to report PQ, which is the main metric for panoptic segmentation, mAP on thing classes for instance segmentation, and mIoU by merging instance masks from the same category for semantic segmentation. Compared to other methods, PSALM achieves comparable performance at similar visual backbone sizes, demonstrating that PSALM is a powerful architecture, even when compared to approaches designed for specific tasks.

Table 9: Comparison with the state-of-the-art methods on Panoptic COCO-val. We abbreviate the datasets: COCO-SAM(CM) [21], VIPSeg (VIP) [29], while others following Tab. 8.

Method	Backbone	Seg. Data	PQ	mAP	mIoU
Mask2Former [6]	Swin-B	C	55.1	45.2	65.1
Mask2Former [6]	Swin-L	C	57.8	48.6	67.4
X-Decoder [60]	DaViT-B	C, L, RC	56.2	45.8	66.0
SEEM [61]	DaViT-B	C, L, RC	56.1	46.4	66.3
OMG-Seg [21]	ConvNeXt-XXL	C, VIP, CM, V	55.4	-	-
PSALM	Swin-B	C, RC, CI	55.9	45.7	66.6

Interactive Segmentation. We also evaluate PSALM in the interactive segmentation tasks. Since the task does not have a well-developed dataset containing all four instructions, previous works have typically used the in-house dataset, thus we re-evaluate other methods on the COCO interactive validation set. The results are shown in Tab. 10, and PSALM achieves leading performance on point, scribble, and mask instructions than all other methods, while on box instruction, SAM performs better on mIoU but worse on cIoU, and we hypothesize might be caused by the different distribution of training data, and the fact that SAM is trained on SA-1B [15], which has a much larger data scale than what we used. In addition, we also report the official results of SEEM, which only evaluated 600 samples from the COCO validation set as a reference.

Table 10: Comparison with the state-of-the-art methods on COCO-Interactive. The results of SEEM-B* is the result reported in official paper, which is evaluated on 600 random samples of `COCO-val`, while all others are evaluated on all samples of `COCO-val`. Abbreviations for each dataset are the same as Tab. 9.

Method	Seg. Data	Point		Scribble		Box		Mask	
		mIoU	cIoU	mIoU	cIoU	mIoU	cIoU	mIoU	cIoU
SAM-B [15]	SA-1B	48.7	33.6	-	-	73.7	68.7	-	-
SAM-L [15]	SA-1B	51.8	37.7	-	-	76.6	71.6	-	-
SEEM-B [61]	C, L, RC	47.8	57.8	43.0	44.0	44.9	42.1	48.4	65.0
SEEM-B* [61]	C, L, RC	81.7	-	83.5	-	75.7	-	76.0	-
OMG-Seg [21]	C, VIP, CM, V	59.3	-	-	-	-	-	-	-
PSALM	C, RC, CI	64.3	74.0	66.9	80.0	67.3	80.9	67.6	82.4

4.3 Generalizability on Out-of-Domain Tasks

Thanks to the flexible design of architecture and input schema, multi-task joint-training, and the strong visual understanding capability of LMM, PSALM shows excellent performance on in-domain tasks, but more importantly, PSALM also demonstrates great potential to generalize to out-of-domain tasks in the zero-shot setting. In this section, we conduct experiments on three different out-of-domain tasks: open-vocabulary segmentation, generalized referring expression segmentation, and video object segmentation. We also tested the zero-shot result of the correspondence benchmark in Ego-Exo4D [12] and Video Object Segmentation in the Appendix.

Table 11: Comparison with the state-of-the-art methods on open-vocabulary instance segmentation and semantic segmentation benchmarks. We use mAP for instance segmentation and mIoU for semantic segmentation.We abbreviate the datasets: Pascal Context-459(PC459) [30], Pascal Context-59(PC59) [30], Pascal VOC-20(PAS20) [11]

Method	OV Instance Seg.		OV Semantic Seg.			
	A150	Cityscapes	PC459	A150	PC59	PAS20
MaskCLIP [9]	6.0	-	10.0	23.7	45.9	-
ODISE [49]	14.4	-	14.5	29.9	57.3	-
SAN [51]	10.6	-	17.1	33.3	60.2	95.5
PSALM	9.0	20.5	10.2	18.2	48.5	81.3
PSALM+LVIS	13.9	19.3	14.0	24.4	57.2	95.0

Open-Vocabulary Segmentation. We first evaluate PSALM in open-vocabulary segmentation tasks, which require the model to have the ability to deal with unseen categories in training. Here, we conduct experiments on both open-vocabulary instance segmentation and open-vocabulary semantic segmentation, and Tab. 11 shows the results. Without any special design, PSALM achieves reasonably good performance, although it is still worse than the best specific method in this task, such as SAN, but we believe that PSALM has a strong potential to be further improved by adding more diverse training data, which is advantages of our method. We have also made a preliminary attempt by further involving the LVIS dataset, and as we expected, the performance has significantly improvements. In addition, existing open-vocabulary segmentation methods are built upon the CLIP model or diffusion model, while our approach is based on LMM models, which is a new path and attempt to bring new inspiration to the community, which we believe is even more important than the performance.

Table 12: Our method's zero-shot performance on gRefCOCO. (ft) denotes models further fine-tuned on gRefCOCO after mix training.

Methods	LLM Type	Zero-Shot	val		testA		testB	
			cIoU	gIoU	cIoU	gIoU	cIoU	gIoU
MattNet [56]	-	✗	47.5	48.2	58.7	59.3	45.3	46.1
LTS [8]	-	✗	52.3	52.7	61.9	62.6	49.9	50.4
ReLA [25]	-	✗	**62.4**	**63.6**	**69.3**	70.0	59.9	61.0
LISA [17]	Vicuna-7B	✗	38.7	32.2	52.6	48.5	44.8	39.7
LISA(ft) [17]	Vicuna-7B	✗	61.7	63.3	69.2	70.1	60.3	61.3
GSVA [48]	Vicuna-7B	✗	61.7	63.3	69.2	**70.1**	**60.3**	**61.3**
GSVA(ft) [48]	Vicuna-7B	✗	63.3	66.5	69.9	71.1	60.5	62.2
PSALM	Phi-1.5 (1.3B)	✓	42.0	43.3	52.4	54.5	50.6	52.5

Generalized Referring Expression Segmentation. The referring segmentation datasets used in training contain only a single object, however, the design of the mask proposal allows PSALM to directly address multi-target without any further training or fine-tuning. We evaluate the gRefCOCO benchmark which contains multiple segment targets. In practice, given an expression, we compute the similarity with all mask proposals and retain all masks with similarity greater than 0.6 as foreground. Tab. 12 shows the results, PSALM also achieved very promising performance, even outperforming the LISA version that only pre-trained on gRefCOCO but without task-specific fine-tuning.

5 Conclusion

The PSALM proposed in this study extends the capability of LMM from text output tasks to image segmentation, addresses the output limitations of the LMM, and unifies various segmentation tasks. PSALM exhibits excellent performance in multiple in-domain tasks, and its generalization ability in out-of-domain tasks further underscores its potential.

Acknowledgments

This work was supported by the National Science Fund for Distinguished Young Scholars of China (Grant No.62225603).

References

1. Alayrac, J.B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., Reynolds, M., et al.: Flamingo: a visual language model for few-shot learning. Advances in Neural Information Processing Systems **35**, 23716–23736 (2022)
2. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners. Advances in neural information processing systems **33**, 1877–1901 (2020)
3. Caesar, H., Uijlings, J., Ferrari, V.: Coco-stuff: Thing and stuff classes in context. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1209–1218 (2018)
4. Chen, K., Zhang, Z., Zeng, W., Zhang, R., Zhu, F., Zhao, R.: Shikra: Unleashing multimodal llm's referential dialogue magic. arXiv preprint arXiv:2306.15195 (2023)
5. Chen, X., Mottaghi, R., Liu, X., Fidler, S., Urtasun, R., Yuille, A.: Detect what you can: Detecting and representing objects using holistic models and body parts. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1971–1978 (2014)
6. Cheng, B., Misra, I., Schwing, A.G., Kirillov, A., Girdhar, R.: Masked-attention mask transformer for universal image segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 1290–1299 (2022)
7. Chiang, W.L., Li, Z., Lin, Z., Sheng, Y., Wu, Z., Zhang, H., Zheng, L., Zhuang, S., Zhuang, Y., Gonzalez, J.E., et al.: Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. See https://vicuna. lmsys. org (accessed 14 April 2023) (2023)
8. Ding, H., Liu, C., Wang, S., Jiang, X.: Vision-language transformer and query generation for referring segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 16321–16330 (2021)
9. Ding, Z., Wang, J., Tu, Z.: Open-vocabulary universal image segmentation with maskclip (2023)
10. Dong, R., Han, C., Peng, Y., Qi, Z., Ge, Z., Yang, J., Zhao, L., Sun, J., Zhou, H., Wei, H., et al.: Dreamllm: Synergistic multimodal comprehension and creation. arXiv preprint arXiv:2309.11499 (2023)
11. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2010 (VOC2010) Results. http://www.pascal-network.org/challenges/VOC/voc2010/workshop/index.html
12. Grauman, K., Westbury, A., Torresani, L., Kitani, K., Malik, J., Afouras, T., Ashutosh, K., Baiyya, V., Bansal, S., Boote, B., et al.: Ego-exo4d: Understanding skilled human activity from first-and third-person perspectives. arXiv preprint arXiv:2311.18259 (2023)
13. Gupta, A., Dollar, P., Girshick, R.: Lvis: A dataset for large vocabulary instance segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 5356–5364 (2019)
14. Jain, J., Li, J., Chiu, M.T., Hassani, A., Orlov, N., Shi, H.: Oneformer: One transformer to rule universal image segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2989–2998 (2023)
15. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., et al.: Segment anything. arXiv preprint arXiv:2304.02643 (2023)

16. Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L.J., Shamma, D.A., et al.: Visual genome: Connecting language and vision using crowdsourced dense image annotations. International journal of computer vision **123**, 32–73 (2017)
17. Lai, X., Tian, Z., Chen, Y., Li, Y., Yuan, Y., Liu, S., Jia, J.: Lisa: Reasoning segmentation via large language model. arXiv preprint arXiv:2308.00692 (2023)
18. Li, B., Zhang, P., Yang, J., Zhang, Y., Pu, F., Liu, Z.: Otterhd: A high-resolution multi-modality model. arXiv preprint arXiv:2311.04219 (2023)
19. Li, J., Li, D., Savarese, S., Hoi, S.: Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. arXiv preprint arXiv:2301.12597 (2023)
20. Li, J., Li, D., Xiong, C., Hoi, S.: Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In: International Conference on Machine Learning. pp. 12888–12900. PMLR (2022)
21. Li, X., Yuan, H., Li, W., Ding, H., Wu, S., Zhang, W., Li, Y., Chen, K., Loy, C.C.: Omg-seg: Is one model good enough for all segmentation? arXiv preprint arXiv:2401.10229 (2024)
22. Li, Y., Bubeck, S., Eldan, R., Del Giorno, A., Gunasekar, S., Lee, Y.T.: Textbooks are all you need ii: phi-1.5 technical report. arXiv preprint arXiv:2309.05463 (2023)
23. Li, Z., Yang, B., Liu, Q., Ma, Z., Zhang, S., Yang, J., Sun, Y., Liu, Y., Bai, X.: Monkey: Image resolution and text label are important things for large multi-modal models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 26763–26773 (2024)
24. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13. pp. 740–755. Springer (2014)
25. Liu, C., Ding, H., Jiang, X.: Gres: Generalized referring expression segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 23592–23601 (2023)
26. Liu, H., Li, C., Li, Y., Li, B., Zhang, Y., Shen, S., Lee, Y.J.: Llava-next: Improved reasoning, ocr, and world knowledge (2024)
27. Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual instruction tuning. Advances in neural information processing systems **36** (2024)
28. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 10012–10022 (2021)
29. Miao, J., Wang, X., Wu, Y., Li, W., Zhang, X., Wei, Y., Yang, Y.: Large-scale video panoptic segmentation in the wild: A benchmark. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 21033–21043 (2022)
30. Mottaghi, R., Chen, X., Liu, X., Cho, N.G., Lee, S.W., Fidler, S., Urtasun, R., Yuille, A.: The role of context for object detection and semantic segmentation in the wild. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2014)
31. Nagaraja, V.K., Morariu, V.I., Davis, L.S.: Modeling context between objects for referring expression understanding. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14. pp. 792–807. Springer (2016)

32. OpenAI: Gpt-4v(ision) system card (2023)
33. Peng, Z., Wang, W., Dong, L., Hao, Y., Huang, S., Ma, S., Wei, F.: Kosmos-2: Grounding multimodal large language models to the world. arXiv preprint arXiv:2306.14824 (2023)
34. Pi, R., Yao, L., Gao, J., Zhang, J., Zhang, T.: Perceptiongpt: Effectively fusing visual perception into llm. arXiv preprint arXiv:2311.06612 (2023)
35. Plummer, B.A., Wang, L., Cervantes, C.M., Caicedo, J.C., Hockenmaier, J., Lazebnik, S.: Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In: Proceedings of the IEEE international conference on computer vision. pp. 2641–2649 (2015)
36. Pont-Tuset, J., Perazzi, F., Caelles, S., Arbeláez, P., Sorkine-Hornung, A., Van Gool, L.: The 2017 davis challenge on video object segmentation. arXiv preprint arXiv:1704.00675 (2017)
37. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PMLR (2021)
38. Ramanathan, V., Kalia, A., Petrovic, V., Wen, Y., Zheng, B., Guo, B., Wang, R., Marquez, A., Kovvuri, R., Kadian, A., et al.: Paco: Parts and attributes of common objects. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7141–7151 (2023)
39. Rasheed, H., Maaz, M., Shaji, S., Shaker, A., Khan, S., Cholakkal, H., Anwer, R.M., Xing, E., Yang, M.H., Khan, F.S.: Glamm: Pixel grounding large multimodal model. arXiv preprint arXiv:2311.03356 (2023)
40. Ren, Z., Huang, Z., Wei, Y., Zhao, Y., Fu, D., Feng, J., Jin, X.: Pixellm: Pixel reasoning with large multimodal model. arXiv preprint arXiv:2312.02228 (2023)
41. Shao, S., Li, Z., Zhang, T., Peng, C., Yu, G., Zhang, X., Li, J., Sun, J.: Objects365: A large-scale, high-quality dataset for object detection. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 8430–8439 (2019)
42. Sudre, C.H., Li, W., Vercauteren, T., Ourselin, S., Jorge Cardoso, M.: Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. In: Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support: Third International Workshop, DLMIA 2017, and 7th International Workshop, ML-CDS 2017, Held in Conjunction with MICCAI 2017, Québec City, QC, Canada, September 14, Proceedings 3. pp. 240–248. Springer (2017)
43. Sun, Q., Yu, Q., Cui, Y., Zhang, F., Zhang, X., Wang, Y., Gao, H., Liu, J., Huang, T., Wang, X.: Generative pretraining in multimodality. arXiv preprint arXiv:2307.05222 (2023)
44. Team, G., Anil, R., Borgeaud, S., Wu, Y., Alayrac, J.B., Yu, J., Soricut, R., Schalkwyk, J., Dai, A.M., Hauth, A., et al.: Gemini: a family of highly capable multimodal models. arXiv preprint arXiv:2312.11805 (2023)
45. Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al.: Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288 (2023)
46. Wang, W., Lv, Q., Yu, W., Hong, W., Qi, J., Wang, Y., Ji, J., Yang, Z., Zhao, L., Song, X., et al.: Cogvlm: Visual expert for pretrained language models. arXiv preprint arXiv:2311.03079 (2023)
47. Wu, J., Jiang, Y., Liu, Q., Yuan, Z., Bai, X., Bai, S.: General object foundation model for images and videos at scale. arXiv preprint arXiv:2312.09158 (2023)

48. Xia, Z., Han, D., Han, Y., Pan, X., Song, S., Huang, G.: Gsva: Generalized segmentation via multimodal large language models. arXiv preprint arXiv:2312.10103 (2023)
49. Xu, J., Liu, S., Vahdat, A., Byeon, W., Wang, X., De Mello, S.: Open-vocabulary panoptic segmentation with text-to-image diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2955–2966 (2023)
50. Xu, J., Xu, L., Yang, Y., Li, X., Xie, Y., Huang, Y.J., Li, Y.: u-llava: Unifying multi-modal tasks via large language model. arXiv preprint arXiv:2311.05348 (2023)
51. Xu, M., Zhang, Z., Wei, F., Hu, H., Bai, X.: Side adapter network for open-vocabulary semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2945–2954 (2023)
52. Xu, M., Zhang, Z., Wei, F., Lin, Y., Cao, Y., Hu, H., Bai, X.: A simple baseline for open-vocabulary semantic segmentation with pre-trained vision-language model. In: European Conference on Computer Vision. pp. 736–753. Springer (2022)
53. Yan, B., Jiang, Y., Wu, J., Wang, D., Luo, P., Yuan, Z., Lu, H.: Universal instance perception as object discovery and retrieval. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15325–15336 (2023)
54. Yang, S., Qu, T., Lai, X., Tian, Z., Peng, B., Liu, S., Jia, J.: An improved baseline for reasoning segmentation with large language model. arXiv preprint arXiv:2312.17240 (2023)
55. You, H., Zhang, H., Gan, Z., Du, X., Zhang, B., Wang, Z., Cao, L., Chang, S.F., Yang, Y.: Ferret: Refer and ground anything anywhere at any granularity. arXiv preprint arXiv:2310.07704 (2023)
56. Yu, L., Lin, Z., Shen, X., Yang, J., Lu, X., Bansal, M., Berg, T.L.: Mattnet: Modular attention network for referring expression comprehension. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1307–1315 (2018)
57. Yu, L., Poirson, P., Yang, S., Berg, A.C., Berg, T.L.: Modeling context in referring expressions. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14. pp. 69–85. Springer (2016)
58. Zhang, A., Zhao, L., Xie, C.W., Zheng, Y., Ji, W., Chua, T.S.: Next-chat: An lmm for chat, detection and segmentation. arXiv preprint arXiv:2311.04498 (2023)
59. Zhou, B., Zhao, H., Puig, X., Xiao, T., Fidler, S., Barriuso, A., Torralba, A.: Semantic understanding of scenes through the ade20k dataset. International Journal of Computer Vision **127**, 302–321 (2019)
60. Zou, X., Dou, Z.Y., Yang, J., Gan, Z., Li, L., Li, C., Dai, X., Behl, H., Wang, J., Yuan, L., et al.: Generalized decoding for pixel, image, and language. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15116–15127 (2023)
61. Zou, X., Yang, J., Zhang, H., Li, F., Li, L., Wang, J., Wang, L., Gao, J., Lee, Y.J.: Segment everything everywhere all at once. Advances in Neural Information Processing Systems **36** (2024)