CLIP as RNN: Segment Countless Visual Concepts without Training Endeavor

Shuyang Sun[1,2*] Runjia Li[1*] Philip Torr[1] Xiuye Gu[2†] Siyang Li[2†]
[1]University of Oxford [2]Google Research
{kevinsun, runjia, phst}@robots.ox.ac.uk {siyang, xiuyegu}google.com
https://torrvision.com/clip_as_rnn/

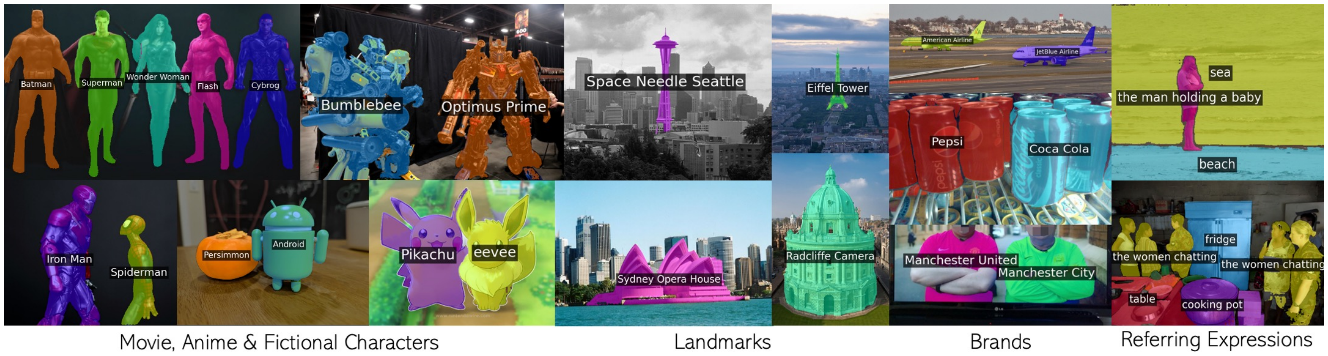

Figure 1. **We propose *CLIP as RNN* (*CaR*) to segment concepts in a vast vocabulary,** including fictional characters, landmarks, brands, everyday objects, and referring expressions. This figure shows our qualitative results. More visualizations are included in the supplementary material. Best viewed in color and with zoom-in.

Abstract

Existing open-vocabulary image segmentation methods require a fine-tuning step on mask labels and/or image-text datasets. Mask labels are labor-intensive, which limits the number of categories in segmentation datasets. Consequently, the vocabulary capacity of pre-trained VLMs is severely reduced after fine-tuning. However, without fine-tuning, VLMs trained under weak image-text supervision tend to make suboptimal mask predictions. To alleviate these issues, we introduce a novel recurrent framework that progressively filters out irrelevant texts and enhances mask quality without training efforts. The recurrent unit is a two-stage segmenter built upon a frozen VLM. Thus, our model retains the VLM's broad vocabulary space and equips it with segmentation ability. Experiments show that our method outperforms not only the training-free counterparts, but also those fine-tuned with millions of data samples, and sets the new state-of-the-art records for both zero-shot semantic and referring segmentation. Concretely, we improve the current record by 28.8, 16.0, and 6.9 mIoU on Pascal VOC, COCO Object, and Pascal Context.

1. Introduction

Natural language serves as a bridge to connect visual elements with human-communicable ideas by transforming colors, shapes, and objects *etc.* into descriptive language. On the other hand, human can use natural language to easily instruct computers and robotics to perform their desired tasks. Built upon the revolutionary vision-language model trained on Internet-scale image-text pairs, *e.g.*, CLIP [49], a variaty of studies [10, 35, 39, 42, 50, 54, 66, 74, 82] have explored using pre-trained VLMs for *open-vocabulary image segmentation* — to segment any concepts in the image described by arbitrary text queries.

Among these advances, several works [35, 40, 74] have integrated pre-trained VLMs with segmenters fine-tuned on bounding boxes and masks. While these methods exhibit superior performances on segmentation benchmarks with common categories, their ability to handle a broader vocabulary is hampered by the small category lists in the segmentation datasets used for fine-tuning. As depicted in Figure 2, even though all three methods incorporate CLIP [49], those relying on fine-tuning with mask annotations [35, 40] fail to recognize the concepts like *Pepsi* and *Coca Cola*.

Since box and mask annotations are expensive, another line of works [10, 39, 42, 50, 51, 66] seek to fine-tune the VLM and/or auxiliary segmentation modules with image-

[*] The first two authors contribute equally to this work. The majority of this work was done during Shuyang's internship at Google Research.
[†] Equal advising.

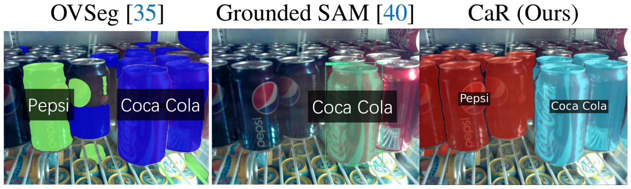

Figure 2. **Our method CaR can fully inherit the vast vocabulary space of CLIP,** by directly using features from a pre-trained VLM, *i.e*., CLIP, without any fine-tuning. Although the scene in the image is simple, state-of-the-art methods fine-tuned on segmentation datasets [35, 40] fail to segment and recognize *Pepsi* and *Coca Cola* correctly.

level annotations only, *e.g*., paired image-text data obtained from the Internet. This would lead to a complicated fine-tuning pipeline. Besides, these segmentation models often have suboptimal mask qualities, as image-level labels cannot directly supervise pixel grouping.

In this paper, we eliminate the fine-tuning on mask annotations or additional image-text pairs to fully preserve the extensive vocabulary space of the pre-trained VLM. However, the pre-training objectives of VLMs are not specifically designed for dense predictions. As a result, existing approaches [14, 37, 82] that do not fine-tune the VLMs struggle to generate accurate visual masks corresponding to the text queries, particularly when some of the text queries refer to non-existing objects in the image. To address this issue, we repeatedly assess the degree of alignment between mask proposals and text queries, and progressively remove text queries with low confidence. As the text queries become cleaner, better mask proposals are consequently obtained. To facilitate this iterative refinement, we propose a novel recurrent architecture with a two-stage segmenter as the recurrent unit, maintaining the same set of weights across all time steps. The two-stage segmenter consists of a mask proposal generator and a mask classifier to assess the mask proposals. Both are built upon a pre-trained CLIP model without modifications. Given an input image and multiple text queries, our model recurrently aligns the visual and textual spaces and generates refined masks as the final output, continuing until a stable state is achieved. Owing to its recurrent nature, we name our entire framework as *CLIP as RNN* (*CaR*).

Experimental results demonstrate our approach is remarkably effective. In comparison with methods that do not use additional training data, *i.e*., zero-shot open-vocabulary semantic segmentation, our approach outperforms the prior art by 28.8, 16.0, and 6.9 mIoU on Pascal VOC [19], COCO Object [36], and Pascal Context [45], respectively. Impressively, even when pitted against models fine-tuned on extensive additional data, our strategy surpasses the best record by 12.6, 4.6, and 0.1 on the three aforementioned datasets, respectively. To assess our model's capacity to handle more complex text queries, we evaluate on the referring image segmentation benchmarks, Ref-COCO, Ref-COCO+ and RefCOCOg. CaR outperforms the zero-shot counterparts by a large margin. Moreover, we extend our method to the video domain, and establish a zero-shot baseline for the video referring segmentation on Ref-DAVIS 2017 [29]. As showcased in Figure 1, our proposed approach CaR exhibits remarkable success across a broad vocabulary spectrum, effectively processing diverse queries from celebrities and landmarks to referring expressions and general objects.

Our contributions can be summarized as follows: 1. By constructing a recurrent architecture, our method CaR performs visual segmentation with arbitrary text queries in a vast vocabulary space without the need of fine-tuning. 2. When compared with previous methods on zero-shot open-vocabulary semantic segmentation and referring segmentation, our method CaR outperforms the prior state of the art by a large margin.

2. Related Work

Open-vocabulary segmentation with mask annotations. The success of VLMs [25, 34, 49, 57, 71, 76, 77] has motivated researchers to push the boundaries of traditional image segmentation tasks, moving them beyond fixed label sets and into an open vocabulary by fine-tuning or training VLMs on segmentation datasets [20, 22, 26, 32, 35, 40, 44, 68, 74, 78, 79, 83]. However, as collecting mask annotations for a vast range of fine-grained labels is prohibitively expensive, existing segmentation datasets, *e.g*. [4, 19, 36, 45, 81] have limited vocabularies. Methods fine-tuned on these mask annotations reduce the open-vocabulary capacity inherited from the pre-trained VLMs. In this work, we attempt to preserve the completeness of the vocabulary space in pre-trained VLMs.

Open-vocabulary segmentation without mask supervision. Several works [6, 10, 11, 23, 42, 46, 50, 51, 54, 66, 67, 82] avoid the aforementioned vocabulary reduction issue by not fine-tuning on any mask annotations. Instead, researchers allow semantic grouping to emerge automatically without any mask supervision. GroupViT [66] learns to progressively group semantic regions with weak supervision, using only image-text datasets. Furthermore, it is possible to use a pre-trained VLM for open-vocabulary segmentation without any additional training [27, 54, 82]. For example, MaskCLIP [82] enables CLIP to perform open-vocabulary segmentation by only modifying its image encoder. However, these methods often suffer from inferior segmentation performance due to the lack of mask supervision, and the modification of the pre-trained VLMs. CaR is closely related to these approaches, we are both in a zero-shot manner without training. CaR stands out by proposing a recurrent framework on a VLM with fixed weights and no alternation on its architecture. Note that our zero-shot is different from the zero-shot semantic segmentation [2, 3, 17, 24, 33, 64, 82] that mirrors the seen/unseen

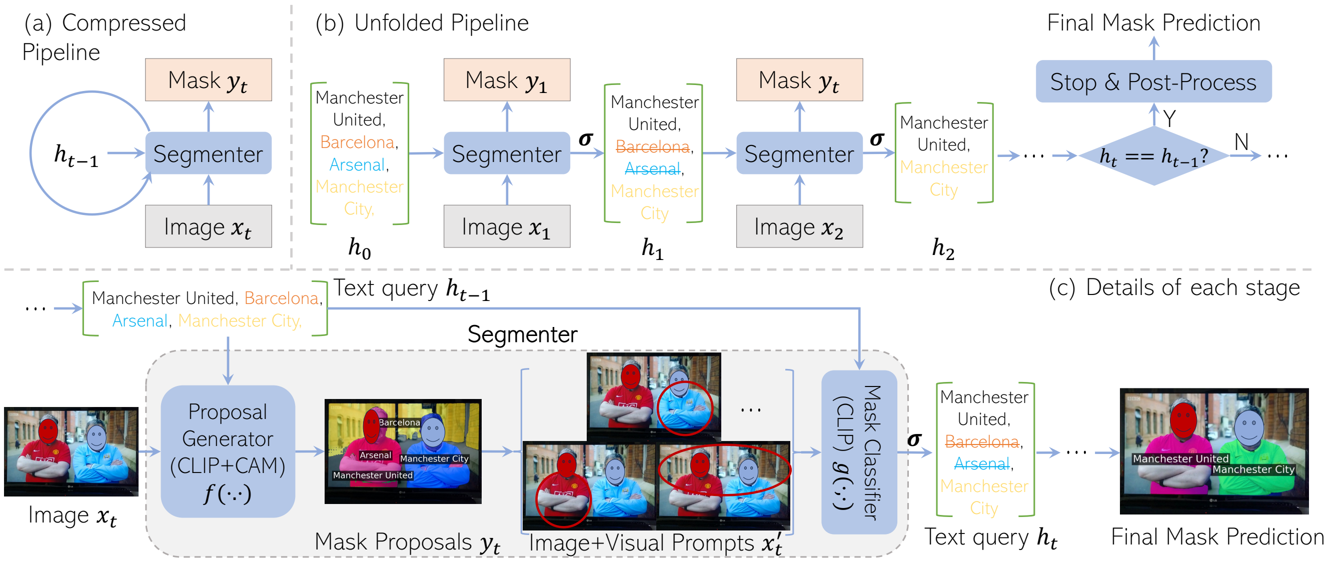

Figure 3. **The overall framework of our method CaR. (a)**, **(b)**: given an image, the user provides a set of text queries that they are interested to segment. This initial set, denoted by h_0, may refer to non-existing concepts in the image, *e.g.*, `Barcelona` and `Arsenal`. In the t-th time step, the frozen segmenter evaluates the degree of alignment between each mask and text query from the previous time step, h_{t-1}, and then low-confidence queries are eliminated by the function σ. **(c)** depicts the detailed architecture of our two-stage segmenter. It consists a mask proposal generator $f(\cdot,\cdot)$, and a mask classifier $g(\cdot,\cdot)$ that assesses the alignment of each mask-text pairs.

class separation from zero-shot classification in earlier ages.

Segmentation with VLM-generated pseudo-labels. As an alternative direction, recent works have exploited pre-trained VLMs to generate pseudo-masks in a fixed label space, requiring only image-level labels or captions for training [1, 37, 41, 52, 54, 65, 69, 82]. Once pseudo mask labels are obtained, a segmenter with a fixed vocabulary (*e.g.*, DeepLab [12, 13]) can be trained in a fully supervised manner. Among these, CLIP-ES [37] is particularly relevant as it directly uses CLIP for pseudo-mask generation given the class names in ground-truth. However, CLIP-ES [37] requires pseudo-label training while we do not.

Progressive refinement for image segmentation. Progressive refinement in image segmentation has seen significant advancements through various approaches. Recent works [8, 15, 16, 58, 60, 73] such as Cascade R-CNN [7], DETR [8] and CRF-RNN [80] combine a detector (R-CNN [21]), a transformer [59] or a segmenter (dense-CRF [31]) repeatedly for iterative refinement. We kindly note that all these works are designed for supervised image instance or semantic segmentation in a closed-set vocabulary. Our method does not require any training effort, yet our way of progressive refinement is fundamentally different from these methods.

3. CLIP as Recurrent Neural Networks

3.1. A Recap on Recurrent Neural Networks

We begin with a concise overview of recurrent neural networks (RNN). RNNs are specifically designed to process sequential data, such as text, speech, and time series. A basic RNN, commonly known as a vanilla RNN, uses the *same* set of weights to process data at all time steps. At each time step t, the process can be expressed as follows:

$$h_t = \sigma(W_{hh}h_{t-1} + W_{xh}x_t + b_h), \tag{1}$$

$$y_t = W_{hy}h_t + b_y. \tag{2}$$

x_t represents the input, and h_t represents the hidden state which serves as the "memory" to store information from previous steps. y_t denotes the output. W_{hh}, W_{xh}, and W_{hy} are weight matrices, b_h and b_y refer to bias terms, and σ denotes a thresholding function to introduce non-linearity.

An RNN's core lies in its hidden state, h_t, which captures information from past time steps. This empowers RNNs to exploit temporal dynamics within sequences. In our approach CaR, we design a similar process - iteratively aligning the textual and visual domains by assessing the accuracy of each text query through a segmenter, using the *same* set of weights as well. The text queries at each step act like the RNN's hidden state, representing the entities identified in the image at each specific time step.

3.2. Overview

As depicted in Figure 3(a) and (b), our training-free framework operates in a recurrent manner, with a fixed-weight segmenter shared across all time steps. In the t-th time step, the segmenter receives an image $x_t \in \mathbb{R}^{3\times H\times W}$ and a set of text queries h_{t-1} from the preceding step as the input. It then produces two outputs: a set of masks

Algorithm 1 Pseudo-code of CLIPasRNN in PyTorch style.

```
# img: the input image with shape (3, H, W)
# h_0: a list of the initial N_0 text queries.
# clip: the CLIP model encoding the image and texts.
# cam: the gradient-based CAM model for mask proposal
    generation.
# eta: a threshold to binarize the masks for visual
    prompting.
# theta: a threshold defined in Eq. 6.

h_{t-1} = h_0
while len(h_{t-1}) > 0:
   # logits: [1, len(h_{t-1})]
   logits = clip(img, h_{t-1})
   scores = softmax(logits, dim=-1)
   # proposals: [len(h_{t-1}), H, W]
   proposals = cam(clip, img, scores)
   # prompted_img: [len(h_{t-1}), H, W]
   prompted_imgs = apply_visual_prompts(img, proposals
      , eta)
   # mask_logits: [len(h_{t-1}), len(h_{t-1})]
   mask_logits = clip(prompted_imgs, h_{t-1})
   mask_scores = softmax(mask_logits, dim=-1)
   # diag_scores: [len(h_{t-1})]
   diag_scores = diagonal(mask_scores)
   h_t = []
   for score, label in zip(diag_scores, h_{t-1}):
      if score > theta:
         h_t.append(label)
   if len(h_t) == len(h_{t-1}):
      break
   h_{t-1} = h_t
final_masks = post_process(proposals)
```

$y_t \in [0,1]^{N_{t-1}\times H\times W}$ corresponding to N_{t-1} input text queries, and the updated text queries h_t for the subsequent step. For image segmentation, all different time steps share the same x_t.

To delve deeper into the design of our framework, we formulate its operations through Eq. (3) to Eq. (5).

$$y_t = f(x_t, h_{t-1}; W_f). \quad (3)$$

Here the function $f(\cdot,\cdot)$ represents the mask proposal generator and W_f denotes its pre-trained weights. The mask proposal generator processes the input image x_t and the text queries at previous step h_{t-1} to generate candidate mask proposals y_t. Given the mask proposal generator is not pre-trained for dense prediction, the mask proposals y_t from $f(\cdot,\cdot)$ are inaccurate. To assess these mask proposals, we draw visual prompts *e.g.*, red circles or background blur, to the input x_t, based on mask proposals to highlight the masked area on the image. The visual prompting function $v(\cdot,\cdot)$ is defined as:

$$x'_t = v(x_t, y_t). \quad (4)$$

Here x'_t represent N_{t-1} images with the visual prompts. The prompted images x'_t are then passed to the mask classifier $g(\cdot,\cdot)$ with the pre-trained weights W_g, along with the text queries h_{t-1}, to compute a similarity matrix P_t. The entire process of the mask classifier can be defined as:

$$P_t = g(x'_t, h_{t-1}; W_g). \quad (5)$$

Finally, after going through a thresholding function $\sigma(\cdot)$, text queries with similarity scores lower than the threshold θ will be removed so that the text queries $h_t = \sigma(P_t)$ for the next step t are obtained. h_t is a potentially reduced set of h_{t-1}. Details of the thresholding function will be given in Section 3.3. This recurrent process continues until the text queries remain unchanged between consecutive steps, *i.e.*, $h_t == h_{t-1}$. We use T to denote this terminal time step. Finally, we apply post-processing described in Section 3.4 to the mask proposals y_T generated in the final time step.

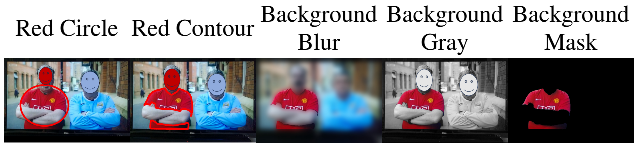

Figure 4. Examples of visual prompts given a mask on the man wearing the jersey of Manchester United.

The pseudo-code in PyTorch-style is given in Algorithm 1. Note that users provide the initial text queries h_0, which are unrestricted and can include general object classes ("cat"), proper nouns ("Space Needle"), referring phrases ("the man in red jacket"), *etc*.

3.3. The Two-stage Segmenter

In this section, we explain the two components of our segmenter, *i.e.* a mask proposal generator and a mask classifier, which serve as the recurrent unit. As illustrated in Figure 3(c), the mask proposal generator first predicts a mask for each text query and then the mask classifier filters out irrelevant text queries based on the degree of alignment with their associated masks. We use the frozen pre-trained CLIP model weights for both the proposal generator and classifier to fully preserve the knowledge encapsulated in CLIP.

Mask proposal generator. To predict the mask proposal y_t, a gradient-based Class-Activation Map (gradCAM) [37, 53] is applied to the pre-trained CLIP. More specifically, the image x_t and text queries h_{t-1} are first fed into CLIP to get a score between the image and each text. We then back-propagate the gradients of the score of each text query (*i.e.*, class) from the feature maps of the CLIP image encoder to obtain a heatmap. Unless otherwise specified, we use the Class Activation Map (CAM) and class affinity (CAA) module from CLIP-ES [37] as our mask proposal generator, with no further training of the CLIP model required. Apart from the text queries at the current step, we explicitly add a set of background queries describing categories that do not exist in the user text queries and calculate their gradients. This helps to suppress the activation from irrelevant texts (*e.g.*, `Barcelona` and `Arsenal` in Figure 3) in the subsequent mask classification process. More details of how CLIP works with gradCAM are provided in the appendix.

Mask classifier. The masks from the proposal generator may be noisy because the input texts are from an unrestricted vocabulary and may refer to non-existing objects in the input image. To remove this type of proposals, we apply another CLIP model to compute a similarity score between each query and its associated mask proposal. A straightforward approach is blacking out all pixels outside the mask region, as shown in the rightmost image in Figure 4, and then computing the visual embedding for the foreground only. However, recent works [41, 55] have found several more effective *visual prompts* which can highlight the foreground as well as preserve the context in the background. Inspired by this, we apply a variety of visual prompts, *e.g.*, red circles, bounding boxes, background blur and gray background to guide the CLIP model to focus on the foreground region. A threshold η is set to first binarize the mask proposals y_t before applying these visual prompts to the images. Please refer to the supplementary material for more implementation details. After applying visual prompts, we obtain N_{t-1} different prompted images, corresponding to N_{t-1} text queries (h_{t-1}). We feed these images and text queries into the CLIP classifier $g(\cdot,\cdot)$ followed by a softmax operation along the text query dimension to get the similarity matrix $P_t \in \mathbb{R}^{N_{t-1}\times N_{t-1}}$ given the image and text embeddings. We only keep the diagonal elements of P_t as the matching score between the i-th mask and the i-th query. If the score is lower than a threshold θ, the query and its mask are filtered out. Mathematically, the thresholding function $\sigma(\cdot)$ is defined as follows:

$$h_t^i = \sigma(P_t^{ii}) = \begin{cases} h_{t-1}^i, & \text{if } P_t^{ii} \geq \theta \\ \texttt{NULL}, & \text{if } P_t^{ii} < \theta \end{cases} \tag{6}$$

where P_t^{ii} is the i-th element of the diagonal of the normalized similarity matrix, and θ is a manually set threshold. NULL represents that the i-th text query is filtered out and will not be input to next steps.

3.4. Post-Processing

Once the recurrent process stops, we start to post-process y_T, the masks from the final step T. We employ dense conditional random field (CRF) [31] to refine mask boundaries. When constructing the CRF, the unary potentials are calculated based on the mask proposals of the last step. All hyper-parameters are set to the defaults in [31]. Finally, an argmax operation is applied to the mask output of denseCRF along the dimension of text queries. Thus, for each spatial location of the mask we only keep the class (text query) with the highest response.

Additionally, we propose to ensemble the CRF-refined masks with SAM [30], as an **optional** post-processing module. This begins with generating a set of mask proposals from SAM using the automask mode, without entering any prompts into SAM. To match these SAM proposals with the masks processed by denseCRF, we introduce a novel metric: the *Intersection over the Minimum-mask (IoM)*. If the IoM between a mask from SAM and a CRF-refined mask surpasses a threshold ϕ_{iom}, we consider them matched. Then all SAM proposals matched to the same CRF-refined mask are combined into one single mask. Finally, we compute the IoU between the combined mask and the original CRF-refined mask. If the IoU is greater than a threshold ϕ_{iou}, we adopt the combined mask to replace the original mask, otherwise, we keep the CRF-refined mask. The detailed post-processing steps are explained in the supplementary material.

4. Experiments

4.1. Zero-shot Semantic Segmentation

Datasets. Since our method does not require training, below we introduce only the datasets utilized for evaluation purposes. We conduct assessments for semantic segmentation using the validation (val) splits of Pascal VOC, Pascal Context, and COCO Object. Specifically, **Pascal VOC** [18] encompasses 21 categories: 20 object classes (VOC-20) alongside one background class (VOC-21). For **Pascal Context** [45], our evaluation employs the prevalent version comprising 59 classes including both "things" and "stuff" categories (PC-59), and one background ("other") class for the concepts outside of the 59 classes (PC-60). Following [66], we construct the **COCO Object** dataset as a derivative of COCO Stuff [5]. We kindly emphasize that the COCO Object dataset is **not** COCO Stuff since it merges all "stuff" classes into one background class and thus has 81 classes (80 "things" + 1 background) in total. We use the standard mean Intersection-over-Union (mIoU) metric to evaluate our method's segmentation performance. Besides, we report the performance of CaR on ADE-150 (A-150), ADE-847 (A-847) and Pascal Context 459 (PC-459) that consist of 150, 847 and 459 classes respectively.

Implementation details. Our proposed method CaR utilizes the foundational pre-trained CLIP models as the backbone. More precisely, we harness the CLIP model with ViT-B/16 to serve as the underlying framework for the mask proposal generator $f(\cdot,\cdot)$. Concurrently, for the mask classifier $g(\cdot,\cdot)$, we adopt a larger ViT-L/14 version for higher precision based on our ablation study. Unless otherwise specified, the reported quantitative results are post-processed solely with a denseCRF, with no SAM masks involved. When setting the threshold hyper-parameters, we assign $\eta = 0.4$, $\theta = 0.6$, and $\lambda = 0.4$ for Pascal VOC, and $\eta = 0.5$, $\theta = 0.3$, $\lambda = 0.5$ for COCO and $\eta = 0.6$, $\theta = 0.2$, $\lambda = 0.4$ for Pascal context. The specific background queries used for the mask generator $f(\cdot,\cdot)$ are ablated in Section 4.2 and detailed in the supplementary material. For Pascal Context, we use separate groups of background queries for "thing" and "stuff". For "thing" categories, all "stuff" categories are added as background queries and vice versa for "stuff" categories. As an optional strategy, we utilize a matching algorithm and perform an ensemble with SAM masks. We set both thresholds, ϕ_{iom} and ϕ_{iou}, to 0.7 for all three datasets. We enable half-precision floating point for CLIP. Since CaR is just a framework de-

Models	Is *VLM* *pre-trained?*	*w/ aux trainable module?*	Additional Training Data	#Images	Additional Supervision	VOC-20‡	VOC-21	COCO Object	PC-59‡	PC-60	A-150	A-847	PC-459
zero-shot methods fine-tuned with additional data													
ViL-Seg [39]	✓	✓	CC12M	12M	text+self	-	34.4	16.4	-	16.3	-	-	-
GroupViT [66]	×	✓	CC12M+YFCC	26M	text	74.1	52.3	24.3	23.4	22.4	10.6	4.3	4.9
GroupViT [66]	×	✓	CC12M+RedCaps	24M	text	79.7	50.8	27.5	-	23.7	-	-	-
SegCLIP [42]	×	✓	CC3M+COCO	3.4M	text+self	-	33.3	15.2	-	19.1	-	-	-
SegCLIP [42]	✓	✓	CC3M+COCO	3.4M	text+self	-	52.6	26.5	-	24.7	8.7	-	-
ZeroSeg [11]	✓	✓	IN-1K	1.3M	self	-	40.8	20.2	-	20.4	-	-	-
ViewCo [51]	✓	✓	CC12M+YFCC	26M	text+self	-	52.4	23.5	-	23.0	-	-	-
MixReorg [6]	✓	✓	CC12M	12M	text	-	47.9	-	-	23.9	-	-	-
CLIPpy [50]	✓	×	HQITP-134M	134M	text+self	-	52.2	32.0	-	-	13.5	-	-
OVSegmenter [67]	✓	✓	CC4M	4M	text	-	53.8	25.1	-	20.4	-	-	-
TCL [10]	✓	✓	CC15M	15M	text+self	77.5	51.2	30.4	24.3	30.3	14.9	-	-
TCL+PAMR [10]	✓	✓	CC15M	15M	text+self	83.2	55.0	31.6	33.9	30.4	17.1	-	-
zero-shot methods with SAM													
SAMCLIP [61] (*w/* [30])	✓	✓	CC15M+YFCC+IN21k	41M	text+self	-	60.6	-	-	29.2	17.1	-	-
CaR+SAM (Ours, *w/* [28])	✓	-	-	-	-	-	**70.2**	**37.6**	-	**31.1**	-	-	-
zero-shot methods without fine-tuning on CLIP													
ReCo† [54]	✓	×	-	-	-	57.7	25.1	15.7	22.3	19.9	11.2	-	-
MaskCLIP† [82]	✓	×	-	-	-	74.9	38.8	20.6	26.4	23.6	9.8	-	-
CaR (Ours, ***w/o* SAM**)	✓	×	-	-	-	**91.4**	**67.6**	**36.6**	**39.5**	**30.5**	**17.7**	**8.1**	**13.9**
Δ *w/ the state-of-the-art w/o additional data*						+16.5	+28.8	+16.0	+13.1	+6.9	+6.5	-	-
Δ *w/ the state-of-the-art w/ additional data*						+8.2	+12.6	+4.6	+5.6	+0.1	+0.6	+3.8	+9.0

Table 1. **Comparison to state-of-the-art zero-shot semantic segmentation approaches.** Results annotated with a † are reported by Cha et al. [10]. A ✓ is placed if either the visual or text encoder of the VLM is pre-trained. The table shows that our method outperforms not only counterparts without fine-tuning by a large margin, but also those fine-tuned on millions of data samples. For fair comparison, we compare with methods using CLIP [49] as the backbone. ‡VOC-20 and PC-59 represent Pascal VOC and Pascal Context **without** background. VOC-21 and PC-60 represent Pascal VOC and Pascal Context with an additional background class. A-150, A-847 and PC-459 represent ADE-150, ADE-847, and Pascal Context 459 datasets, respectively.

Dataset	*w/ recurrence?*	CAM	mIoU
Pascal VOC		CLIP-ES [37]	15.2
	✓	CLIP-ES [37]	67.6
	✓	gradCAM [53]	41.1

Table 2. **Effect of applying our recurrent architecure and different CAM methods.** The recurrence plays a vital role in improving the performance.

signed for inference, all experiments in this paper are conducted on just **one** NVIDIA V100 GPU.

Efficiency. CaR **without SAM** operates at 950 ms with CPU-based CRF (200ms) for 500 × 500 images. GPU CRF is $\sim 5\times$ faster. CaR is memory-efficient, consuming only 3.6GB GPU memory on Pascal VOC.

CaR significantly outperforms methods without additional training. We also compare CaR with training-free methods like MaskCLIP [82] and ReCo [54]. Across the benchmarks, our model consistently demonstrates an impressive performance uplift. Under a similar setting with no additional training data, CaR surpasses previous state-of-the-art method by 28.8, 16.0 and 6.9 mIoU on Pascal VOC, COCO Object and Pascal Context, respectively.

Training-free CaR even outperforms several methods with additional fine-tuning. As shown in Table 1, we compare our method with previous state-of-the-art methods including ViL-Seg [39], GroupViT [66], SegCLIP [42], ZeroSeg [11], ViewCo [51], CLIPpy [50], and TCL [10], which are augmented with additional data. The prior best results of different datasets are achieved by different methods. Specifically, TCL [10], employing a fully pre-trained CLIP model and fine-tuned on 15M additional data, achieves the highest mIoU (55.0 and 30.4) on Pascal VOC and Pascal Context. CLIPpy [50] sets the previous highest record on COCO Object but also requires extensive data for fine-tuning. Concretely, it first utilizes a ViT-based image encoder pre-trained with DINO [9] and a pre-trained T5 text encoder [48], and then fine-tunes both encoders with 134M additional data. Our method, incurring no cost for fine-tuning, still outperforms these methods by 12.6, 4.5, and 0.1 mIoU on the Pascal VOC, COCO Object, and Pascal Context datasets, respectively. Since "stuff" appears less frequently in the pre-training image-text data for CLIP, CaR also exhibits less sensitivity to "stuff" on Pascal Context.

CaR+SAM further boosts the performance. When integrated with SAM [28, 30], we compare CaR with a concurrent method SAMCLIP [61] and outperform it by 9.6 and 1.9 on Pascal VOC and Pascal Context. We use the recent variant HQ-SAM [28] with **no prompt given** (`automask` mode), and then match the generated masks with metrics designed in Section 3.4. In other words, SAM is only used as a post-processor to refine the boundary of results from

Mask Proposal Generator $f(\cdot,\cdot)$	Mask Classifier $g(\cdot,\cdot)$	Pascal VOC	COCO Object
ViT-B/16	ViT-B/16	54.1	15.9
	ViT-L/14	**67.6**	**36.6**
ViT-L/14	ViT-B/16	50.6	14.1
	ViT-L/14	57.6	32.5

Table 3. **Effect of CLIP backbones.** We compare various CLIP backbones on Pascal VOC and COCO Object. Results show that we can improve the performance by scaling up the mask classifier.

Dataset	Visual Prompts					mIoU
	`circle`	`contour`	`blur`	`gray`	`mask`	
Pascal VOC	✓					66.9
		✓				66.0
			✓			66.4
				✓		66.1
					✓	61.8
	✓		✓			**67.6**
	✓			✓		67.1
		✓	✓			66.5
			✓	✓		66.3
	✓		✓	✓		66.8

Table 4. **Effect of different visual prompts.** When multiple visual prompts are checked, we will apply all checked visual prompts simultaneously on one image. The experiments are conducted on Pascal VOC. Results for COCO and Pascal Context are shown in supplementary materials.

Pascal VOC				COCO Object			
η	θ	λ	mIoU	η	θ	λ	mIoU
0.3	0.6	0.4	67.0	0.5	0.3	0.6	35.4
0.4	0.6	0.4	**67.6**	0.5	0.3	0.4	36.1
0.5	0.6	0.4	67.0	0.4	0.3	0.5	35.8
0.4	0.5	0.4	67.4	0.5	0.3	0.5	**36.6**
0.4	0.7	0.4	67.5	0.6	0.3	0.5	35.9
0.4	0.6	0.3	67.3	0.5	0.4	0.5	36.3
0.4	0.6	0.5	67.0	0.5	0.5	0.5	36.0

Table 5. **Effect of different hyper-parameters**: the threshold to binarize mask proposals (η), the threshold to remove text queries (θ), and parameter of CLIP-ES's[37] (λ). Experiments are conducted on Pascal VOC and COCO Object.

CaR. By applying SAM into our framework, our results can be further boosted by 2.6, 1.1 and 0.6 mIoU on Pascal VOC, COCO Object and Pascal Context, respectively.

4.2. Ablation Studies.

Effect of Recurrence. As illustrated in Table 2, the incorporation of the recurrent architecture is crucial to our method. Without recurrence (*a.k.a* $T = 1$), our method functions similarly to CLIP-ES [37] with an additional CLIP classifier, and achieves only 15.2% in mIoU. The recurrent framework can lead to a 52.4% improvement, reaching an mIoU of 67.6%. The significant improvement validates the effectiveness of the recurrent design of our framework. For VOC and COCO, most images require two steps, and a small portion of images goes beyond two.

Effect of different CAM methods. Table 2 exhibits that our framework is compatible with different CAM methods and could be potentially integrated with other CAM-related designs. When integrated with CLIP-ES [37], our method is 26.5 mIoU higher than that with gradCAM [53]. We kindly note that we do not carefully search the hyper-parameters on gradCAM so the performance could be further improved.

Effect of different CLIP Backbones. We experiment with different settings of CLIP backbones used in the mask proposal generator f and mask classifier g, on Pascal VOC and COCO Object datasets. Results are displayed in Table 3. As for the mask proposal generator, ViT-B/16 outperforms the ViT-L/14 by over 10 mIoU on both Pascal VOC and COCO Object. There is significant mIoU gains when employing the larger ViT-L/14 for the mask classifier over ViT-B/16. Similar observations have been found by Shtedritski et al. [55] that a larger backbone can better understand the visual prompts, which indicates that the performance of our method can be potentially improved by employing large backbones for the mask classifier.

Effect of different visual prompts. There are various forms of visual prompts, including circle, contour, background blur (`blur`), background gray (`gray`), and background mask (`mask`), *etc.* We study the effects of different visual prompts on the Pascal VOC dataset and Table 4 summarizes the results when applying one or a combination of two of the aforementioned visual prompt types. The highest mIoU score is achieved with the combination of `circle` and `blur`, yielding a mIoU of 67.6. Notably, using `mask` alone results in the lowest mIoU of 61.8, which is a common practice for most previous open-vocabulary segmentation approaches, *e.g.*, [35, 74]. We also evaluate the effect of different visual prompts on COCO Object and Pascal Context, and show the results in the supplementary material.

Effect of hyper-parameters. We perform an ablation study on the performance impact of various hyper-parameter configurations on Pascal VOC, and present the results in Table 5. Hyper-parameters include the mask binarization threshold, η, defined in Section 3.3, the threshold θ employed in the thresholding function defined in Eq. (6), and the parameter λ defined in CLIP-ES [37]. The peak performance is recorded at an mIoU of 67.6 for $\eta = 0.4$, $\theta = 0.6$, and $\lambda = 0.4$ on Pascal VOC and 36.6 for $\eta = 0.5$, $\theta = 0.3$, and $\lambda = 0.5$ on COCO Object. Different parameter combinations result in mIoU scores that range from 67.0 to 67.6 on Pascal VOC and from 35.4 to 36.6 on COCO Object.

Effect of background queries. In Table 6, we explore how different background queries (classes not existing in the input queries) can affect CaR's performance. We find that the segmentation quality improves as we include more diverse

Dataset	Background queries			mIoU
	`Terrestrial`	`Aquatic` `Atmospheric`	`Man-Made`	
Pascal VOC	×	×	×	64.3
	✓	×	×	65.6
	×	✓	×	64.9
	×	×	✓	66.4
	✓	✓	×	65.8
	×	✓	✓	66.4
	✓	×	✓	65.8
	✓	✓	✓	**67.6**

Table 6. **Effect of background queries on Pascal VOC.** We divide background queries into: `Terrestrial`, `Aquatic`, `Atmospheric`, and `Man-Made`. We use "None" as the background query for the result in the first row. Specific background queries of each category are shown in the supplementary material.

Models	RefCOCO			RefCOCO+			RefCOCOg			GRES
	val	testA	testB	val	testA	testB	val	test(U)	val(G)	
weakly-supervised										
TSEG [56]	25.95	-	-	22.62	-	-	23.41	-	-	-
zero-shot										
GL CLIP [75]	26.20	24.94	26.56	27.80	25.64	27.84	33.52	33.67	33.61	-
CaR(Ours)	33.57	35.36	30.51	34.22	36.03	31.02	36.67	36.57	36.63	16.8

Table 7. **Comparison to state-of-the-art methods on referring image segmentation in mIoU.** CaR is better than all comparison methods in all benchmarks.

background queries: The combination of all three types of background queries delivers the highest mIoU of 67.6. For more details about the background queries of each class, please refer to the supplementary material.

4.3. Referring Segmentation

We evaluate CaR on the referring segmentation task for both images and videos. Again, our method is an inference-only pipeline built upon pre-trained CLIP models, and does not need training/fine-tuning on any types of annotations. For referring segmentation we only use denseCRF [31] for post-processing, and SAM is **not** involved for all experiments in this section for fair comparison. Please refer to the supplementary material for the implementation details.

Datasets. Following [70, 75], we evaluate on **RefCOCO** [72], **RefCOCO+** [72], **RefCOCOg** [43, 47] and **GRES** [38] for the referring image segmentation task. Images used in all three datasets are sourced from the MS COCO [36] dataset and the masks are paired with descriptive language annotations. In RefCOCO+, descriptions about location are prohibited, making the task more challenging. There are two separate splits of the RefCOCOg dataset, one by UMD (U) [47] and another by Google (G) [63]. Following previous work, we use the standard mIoU metric. Apart from referring image segmentation, we also set up a new baseline for **zero-shot referring video segmentation** on **Ref-DAVIS 2017** [29]. Following [29], we adopt region similarity $\mathcal{J}$, contour accuracy $\mathcal{F}$, and the averaged score $\mathcal{J}\&\mathcal{F}$ as the metrics for evaluation.

$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$
30.34	28.15	32.53

Table 8. **Results on Ref-DAVIS 2017.**

Experimental results. Table 7 compares the performance of CaR with other methods on the referring image segmentation tasks across RefCOCO, RefCOCO+, and RefCOCOg. Comparing with other zero-shot methods, our method CaR outperforms Global-Local CLIP (GL CLIP) on all splits of these benchmarks. The performance gap is most pronounced on RefCOCO's `testA` split, where CaR outperforms 10.42 mIoU, and similarly on RefCOCO+'s `testA` split, with a lead of 10.72 mIoU. We also note that GL CLIP [75] uses a pre-trained segmenter Free-SOLO [62] for mask extraction, while CaR is built **without** any pre-trained segmenter. As the first zero-shot method on GRES [38], CaR achieves 16.8 mIoU with no training effort incurred.

For referring video segmentation, we demonstrate in Table 8 that our method achieves 30.34, 28.15 and 32.53 for $\mathcal{J}\&\mathcal{F}$, $\mathcal{J}$ and $\mathcal{F}$ on Ref-DAVIS 2017 [29]. Considering that our method CaR requires neither fine-tuning nor annotations and operates in a zero-shot manner, this performance establishes a strong baseline.

5. Conclusion

We introduce CLIP as RNN (CaR), which preserves the entire large vocabulary space of pre-trained VLMs, by eliminating the fine-tuning process. By constructing a recurrent pipeline with a shared segmenter in the loop, CaR can perform zero-shot semantic and referring segmentation without any additional training efforts. Experiments show that CaR outperforms previous state-of-the-art counterparts by a large margin on eight different benchmarks, *i.e.* Pascal VOC, COCO Object, Pascal Context, ADE-150, ADE-847 and Pascal Context 459 on zero-shot semantic segmentation. We also demonstrate that CaR can handle referring expressions and segment fine-grained concepts like anime characters and landmarks, and also achieves state-of-the-art performance on RefCOCO, RefCOCO+, RefCOCOg and GRES for zero-shot referring segmentation. We hope our work sheds light on future research in open-vocabulary segmentation aiming to further expand the vocabulary space.

Acknowledgement. The majority of this work was done during Shuyang's internship at Google Research. We would like to thank Jindong Gu and Jianhao Yuan at Oxford University, Anurag Arnab, Xingyi Zhou, Huizhong Chen and Neil Alldrin at Google Research for their insightful discussion, Zhongli Ding for image donation. Shuyang Sun and Philip Torr are supported by UKRI grants: Turing AI Fellowship EP/W002981/1. We would also like to thank the Royal Academy of Engineering and FiveAI.

References

[1] Nikita Araslanov and Stefan Roth. Single-stage semantic segmentation from image labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4253–4262, 2020. 3

[2] Donghyeon Baek, Youngmin Oh, and Bumsub Ham. Exploiting a joint embedding space for generalized zero-shot semantic segmentation. In *ICCV*, 2021. 2

[3] Maxime Bucher, Tuan-Hung Vu, Matthieu Cord, and Patrick Pérez. Zero-shot semantic segmentation. *Advances in Neural Information Processing Systems*, 32, 2019. 2

[4] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Cocostuff: Thing and stuff classes in context. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018. 2

[5] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Cocostuff: Thing and stuff classes in context. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1209–1218, 2018. 5

[6] Kaixin Cai, Pengzhen Ren, Yi Zhu, Hang Xu, Jianzhuang Liu, Changlin Li, Guangrun Wang, and Xiaodan Liang. Mixreorg: Cross-modal mixed patch reorganization is a good mask learner for open-world semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1196–1205, 2023. 2, 6

[7] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *CVPR*, 2018. 3

[8] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020. 3

[9] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021. 6

[10] Junbum Cha, Jonghwan Mun, and Byungseok Roh. Learning to generate text-grounded mask for open-world semantic segmentation from only image-text pairs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11165–11174, 2023. 1, 2, 6

[11] Jun Chen, Deyao Zhu, Guocheng Qian, Bernard Ghanem, Zhicheng Yan, Chenchen Zhu, Fanyi Xiao, Mohamed Elhoseiny, and Sean Chang Culatana. Exploring open-vocabulary semantic segmentation without human labels. *arXiv preprint arXiv:2306.00450*, 2023. 2, 6

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

[13] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, 2018. 3

[14] Peijie Chen, Qi Li, Saad Biaz, Trung Bui, and Anh Nguyen. gscorecam: What objects is clip looking at? In *Proceedings of the Asian Conference on Computer Vision*, pages 1959–1975, 2022. 2

[15] Bowen Cheng, Alexander G Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. In *NeurIPS*, 2021. 3

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

[17] Jian Ding, Nan Xue, Gui-Song Xia, and Dengxin Dai. Decoupling zero-shot semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11583–11592, 2022. 2

[18] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 88:303–338, 2010. 5

[19] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88:303–338, 2010. 2

[20] Golnaz Ghiasi, Xiuye Gu, Yin Cui, and Tsung-Yi Lin. Scaling open-vocabulary image segmentation with image-level labels. In *European Conference on Computer Vision*, pages 540–557. Springer, 2022. 2

[21] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014. 3

[22] Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. Open-vocabulary object detection via vision and language knowledge distillation. *arXiv preprint arXiv:2104.13921*, 2021. 2

[23] Wenbin He, Suphanut Jamonnak, Liang Gou, and Liu Ren. Clip-s4: Language-guided self-supervised semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11207–11216, 2023. 2

[24] Ping Hu, Stan Sclaroff, and Kate Saenko. Uncertainty-aware learning for zero-shot semantic segmentation. *Advances in Neural Information Processing Systems*, 33:21713–21724, 2020. 2

[25] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International conference on machine learning*, pages 4904–4916. PMLR, 2021. 2

[26] Aishwarya Kamath, Mannat Singh, Yann LeCun, Gabriel Synnaeve, Ishan Misra, and Nicolas Carion. Mdetr-modulated detection for end-to-end multi-modal understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1780–1790, 2021. 2

[27] Laurynas Karazija, Iro Laina, Andrea Vedaldi, and Christian Rupprecht. Diffusion models for zero-shot open-vocabulary segmentation. *arXiv preprint arXiv:2306.09316*, 2023. 2

[28] Lei Ke, Mingqiao Ye, Martin Danelljan, Yifan Liu, Yu-Wing Tai, Chi-Keung Tang, and Fisher Yu. Segment anything in high quality. *arXiv preprint arXiv:2306.01567*, 2023. 6

[29] Anna Khoreva, Anna Rohrbach, and Bernt Schiele. Video object segmentation with language referring expressions. In

Computer Vision–ACCV 2018: 14th Asian Conference on Computer Vision, Perth, Australia, December 2–6, 2018, Revised Selected Papers, Part IV 14, pages 123–141. Springer, 2019. 2, 8

[30] 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. *arXiv preprint arXiv:2304.02643*, 2023. 5, 6

[31] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. *Advances in neural information processing systems*, 24, 2011. 3, 5, 8

[32] Boyi Li, Kilian Q Weinberger, Serge Belongie, Vladlen Koltun, and Rene Ranftl. Language-driven semantic segmentation. In *International Conference on Learning Representations*, 2022. 2

[33] Peike Li, Yunchao Wei, and Yi Yang. Consistent structural relation learning for zero-shot segmentation. *NeurIPS*, 2020. 2

[34] Yanghao Li, Haoqi Fan, Ronghang Hu, Christoph Feichtenhofer, and Kaiming He. Scaling language-image pre-training via masking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23390–23400, 2023. 2

[35] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7061–7070, 2023. 1, 2, 7

[36] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 2, 8

[37] Yuqi Lin, Minghao Chen, Wenxiao Wang, Boxi Wu, Ke Li, Binbin Lin, Haifeng Liu, and Xiaofei He. Clip is also an efficient segmenter: A text-driven approach for weakly supervised semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15305–15314, 2023. 2, 3, 4, 6, 7

[38] Chang Liu, Henghui Ding, and Xudong Jiang. GRES: Generalized referring expression segmentation. In *CVPR*, 2023. 8

[39] Quande Liu, Youpeng Wen, Jianhua Han, Chunjing Xu, Hang Xu, and Xiaodan Liang. Open-world semantic segmentation via contrasting and clustering vision-language embedding. In *European Conference on Computer Vision*, pages 275–292. Springer, 2022. 1, 6

[40] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023. 1, 2

[41] Timo Lüddecke and Alexander Ecker. Image segmentation using text and image prompts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7086–7096, 2022. 3, 5

[42] Huaishao Luo, Junwei Bao, Youzheng Wu, Xiaodong He, and Tianrui Li. Segclip: Patch aggregation with learnable centers for open-vocabulary semantic segmentation. In *International Conference on Machine Learning*, pages 23033–23044. PMLR, 2023. 1, 2, 6

[43] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 11–20, 2016. 8

[44] M Minderer, A Gritsenko, A Stone, M Neumann, D Weissenborn, A Dosovitskiy, A Mahendran, A Arnab, M Dehghani, Z Shen, et al. Simple open-vocabulary object detection with vision transformers. arxiv 2022. *arXiv preprint arXiv:2205.06230*, 2022. 2

[45] Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, and Alan Yuille. The role of context for object detection and semantic segmentation in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 2, 5

[46] Jishnu Mukhoti, Tsung-Yu Lin, Omid Poursaeed, Rui Wang, Ashish Shah, Philip HS Torr, and Ser-Nam Lim. Open vocabulary semantic segmentation with patch aligned contrastive learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19413–19423, 2023. 2

[47] Varun K Nagaraja, Vlad I Morariu, and Larry S Davis. 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*, pages 792–807. Springer, 2016. 8

[48] Jianmo Ni, Gustavo Hernández Ábrego, Noah Constant, Ji Ma, Keith B Hall, Daniel Cer, and Yinfei Yang. Sentence-t5: Scalable sentence encoders from pre-trained text-to-text models. *arXiv preprint arXiv:2108.08877*, 2021. 6

[49] 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 *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 1, 2, 6

[50] Kanchana Ranasinghe, Brandon McKinzie, Sachin Ravi, Yinfei Yang, Alexander Toshev, and Jonathon Shlens. Perceptual grouping in vision-language models. *arXiv preprint arXiv:2210.09996*, 2022. 1, 2, 6

[51] Pengzhen Ren, Changlin Li, Hang Xu, Yi Zhu, Guangrun Wang, Jianzhuang Liu, Xiaojun Chang, and Xiaodan Liang. Viewco: Discovering text-supervised segmentation masks via multi-view semantic consistency. *arXiv preprint arXiv:2302.10307*, 2023. 1, 2, 6

[52] Lixiang Ru, Yibing Zhan, Baosheng Yu, and Bo Du. Learning affinity from attention: End-to-end weakly-supervised semantic segmentation with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16846–16855, 2022. 3

[53] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra.

Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017. 4, 6, 7

[54] Gyungin Shin, Weidi Xie, and Samuel Albanie. Reco: Retrieve and co-segment for zero-shot transfer. *Advances in Neural Information Processing Systems*, 35:33754–33767, 2022. 1, 2, 3, 6

[55] Aleksandar Shtedritski, Christian Rupprecht, and Andrea Vedaldi. What does clip know about a red circle? visual prompt engineering for vlms. *arXiv preprint arXiv:2304.06712*, 2023. 5, 7

[56] Robin Strudel, Ivan Laptev, and Cordelia Schmid. Weakly-supervised segmentation of referring expressions. *arXiv preprint arXiv:2205.04725*, 2022. 8

[57] Quan Sun, Yuxin Fang, Ledell Wu, Xinlong Wang, and Yue Cao. Eva-clip: Improved training techniques for clip at scale. *arXiv preprint arXiv:2303.15389*, 2023. 2

[58] Shuyang Sun, Weijun Wang, Qihang Yu, Andrew Howard, Philip Torr, and Liang-Chieh Chen. Remax: Relaxing for better training on efficient panoptic segmentation. *arXiv preprint arXiv:2306.17319*, 2023. 3

[59] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 3

[60] Huiyu Wang, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. Max-deeplab: End-to-end panoptic segmentation with mask transformers. In *CVPR*, 2021. 3

[61] Haoxiang Wang, Pavan Kumar Anasosalu Vasu, Fartash Faghri, Raviteja Vemulapalli, Mehrdad Farajtabar, Sachin Mehta, Mohammad Rastegari, Oncel Tuzel, and Hadi Pouransari. Sam-clip: Merging vision foundation models towards semantic and spatial understanding. *arXiv preprint arXiv:2310.15308*, 2023. 6

[62] Xinlong Wang, Zhiding Yu, Shalini De Mello, Jan Kautz, Anima Anandkumar, Chunhua Shen, and Jose M Alvarez. Freesolo: Learning to segment objects without annotations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14176–14186, 2022. 8

[63] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv:1609.08144*, 2016. 8

[64] Yongqin Xian, Subhabrata Choudhury, Yang He, Bernt Schiele, and Zeynep Akata. Semantic projection network for zero-and few-label semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8256–8265, 2019. 2

[65] Jinheng Xie, Xianxu Hou, Kai Ye, and Linlin Shen. Clims: Cross language image matching for weakly supervised semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4483–4492, 2022. 3

[66] Jiarui Xu, Shalini De Mello, Sifei Liu, Wonmin Byeon, Thomas Breuel, Jan Kautz, and Xiaolong Wang. Groupvit: Semantic segmentation emerges from text supervision. In *CVPR*, 2022. 1, 2, 5, 6

[67] Jilan Xu, Junlin Hou, Yuejie Zhang, Rui Feng, Yi Wang, Yu Qiao, and Weidi Xie. Learning open-vocabulary semantic segmentation models from natural language supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2935–2944, 2023. 2, 6

[68] Jiarui Xu, Sifei Liu, Arash Vahdat, Wonmin Byeon, Xiaolong Wang, and Shalini De Mello. Open-vocabulary panoptic segmentation with text-to-image diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2955–2966, 2023. 2

[69] Lian Xu, Wanli Ouyang, Mohammed Bennamoun, Farid Boussaid, and Dan Xu. Multi-class token transformer for weakly supervised semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4310–4319, 2022. 3

[70] Zhao Yang, Jiaqi Wang, Yansong Tang, Kai Chen, Hengshuang Zhao, and Philip HS Torr. Lavt: Language-aware vision transformer for referring image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18155–18165, 2022. 8

[71] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models. *arXiv preprint arXiv:2205.01917*, 2022. 2

[72] Licheng Yu, Patrick Poirson, Shan Yang, Alexander C Berg, and Tamara L Berg. Modeling context in referring expressions. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*, pages 69–85. Springer, 2016. 8

[73] Qihang Yu, Huiyu Wang, Siyuan Qiao, Maxwell Collins, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. k-means Mask Transformer. In *ECCV*, 2022. 3

[74] Qihang Yu, Ju He, Xueqing Deng, Xiaohui Shen, and Liang-Chieh Chen. Convolutions die hard: Open-vocabulary segmentation with single frozen convolutional clip. *arXiv preprint arXiv:2308.02487*, 2023. 1, 2, 7

[75] Seonghoon Yu, Paul Hongsuck Seo, and Jeany Son. Zero-shot referring image segmentation with global-local context features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19456–19465, 2023. 8

[76] Xiaohua Zhai, Xiao Wang, Basil Mustafa, Andreas Steiner, Daniel Keysers, Alexander Kolesnikov, and Lucas Beyer. Lit: Zero-shot transfer with locked-image text tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18123–18133, 2022. 2

[77] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. *arXiv preprint arXiv:2303.15343*, 2023. 2

[78] Haotian Zhang, Pengchuan Zhang, Xiaowei Hu, Yen-Chun Chen, Liunian Li, Xiyang Dai, Lijuan Wang, Lu Yuan, Jenq-Neng Hwang, and Jianfeng Gao. Glipv2: Unifying localization and vision-language understanding. *Advances in Neural Information Processing Systems*, 35:36067–36080, 2022. 2

[79] Hao Zhang, Feng Li, Xueyan Zou, Shilong Liu, Chunyuan Li, Jianwei Yang, and Lei Zhang. A simple framework for open-vocabulary segmentation and detection. In *Proceed-*

ings of the IEEE/CVF International Conference on Computer Vision, pages 1020–1031, 2023. 2

[80] Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip HS Torr. Conditional random fields as recurrent neural networks. In *ICCV*, 2015. 3

[81] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *International Journal of Computer Vision*, 2019. 2

[82] Chong Zhou, Chen Change Loy, and Bo Dai. Extract free dense labels from clip. In *European Conference on Computer Vision*, pages 696–712. Springer, 2022. 1, 2, 3, 6

[83] Xingyi Zhou, Rohit Girdhar, Armand Joulin, Philipp Krähenbühl, and Ishan Misra. Detecting twenty-thousand classes using image-level supervision. In *European Conference on Computer Vision*, pages 350–368. Springer, 2022. 2