Supplementary Materials for 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/

Appendix

1. More Experimental Results

1.1. Quantitative Analysis on Vocabulary Space.

We demonstrate that our method CaR has a larger vocabulary space compared to the methods fine-tuned with mask annotations. Here we compare our method with OVSeg [5], which is fine-tuned on ImageNet [14] and COCO [6] with a pre-trained CLIP backbone for the task of referring image segmentation. We believe that referring expressions (*e.g.*, "the person in the red shirt" or "the cat in the mirror") refers to a specific segment using a broad vocabulary. We conduct a comparative analysis between a robust open-vocabulary segmentation benchmark, OVSeg [5], and CaR, utilizing standard referring image segmentation benchmarks [9, 12, 19]. We note that RefCOCO and COCO share the same set of images so OVSeg fine-tuning on COCO may not be counted as zero-shot on RefCOCO. The results, as detailed in Table A, demonstrate that CaR significantly surpasses OVSeg in performance. This disparity in performance suggests that CaR encompasses a more expansive vocabulary space than OVSeg.

Models	RefCOCO			RefCOCO+			RefCOCOg		
	val	testA	testB	val	testA	testB	val	test(U)	val(G)
OVSeg [5]	22.58	19.38	25.63	19.13	15.74	25.30	27.87	29.09	28.31
CaR(Ours)	33.57	35.36	30.51	34.22	36.03	31.02	36.67	36.57	36.63

Table A. **Comparison to mask-supervised open-vocabulary methods on referring image segmentation in mIoU.** CaR is better than the comparison method, OVSeg, in all splits of the three benchmarks.

1.2. Evaluation without Background

Following [7], our methodology benefits from using background queries in CLIP [13] classification to suppress false positives (predictions not belonging to the input text queries), enhancing segmentation results. Nevertheless, for more comprehensive comparison, we also assess our approach using an alternate evaluation setting, previously established, which omits the background class. Consequently, less emphasis is placed on object boundaries in this setting. We test our method on two datasets: Pascal VOC [3] without background (termed *VOC-20*) and Pascal Context [10] without background (termed *Context-59*). This setting tests the ability of various methods to discriminate between different classes. Our method CaR significantly outperforms previous methods on VOC-20 and Context-59, where all methods use the same setting that ignores the background class. We reached out to the PACL authors to confirm that they did not evaluate background.

Model	Is VLM pre-trained?	#Additional Images	*w/o* Background	
			VOC-20	Context-59
GroupViT† [18]	×	26M	79.7	23.4
PACL [11]	✓	40M	72.3	50.1
TCL [2]	✓	15M	77.5	30.3
MaskCLIP† [20]	✓	-	74.9	26.4
ReCo† [16]	✓	-	57.5	22.3
CaR (Ours)	✓	-	**91.4**	**39.5**

Table B. **Comparison with methods under the setting where background is ignored.** We compare CaR with prior work on VOC-20, Context-59 in a setting that considers only the foreground pixels (decided by ground truth). Our method shows comparable performance to prior works despite only relying on pre-trained feature extractors. †: numbers are from [2].

2. Implementation Details of CAM

In this paper, we integrate two kinds of gradient-based CAM, *i.e.*, Grad-CAM [15] and CLIP-ES [7], respectively.

Integration with Grad-CAM. When integrating Grad-CAM [15] into our framework, we first extract the image and text feature vectors $v_x = f_I(x), v_h = f_T(h)$ from the image and text encoder $f_I(\cdot), f_T(\cdot)$ given an image x and text queries h. We compute a similarity score between the image and text features using the dot product

$s = \text{softmax}(v_x \cdot v_h^\intercal)$, where softmax is applied along the dimension of v_h. This score s quantifies the alignment (*a.k.a* similarity) between the image x and the text h as perceived by the CLIP model. Here h contains multiple queries. To integrate Grad-CAM into our framework, we first compute the gradients of the similarity score with respect to the feature maps of the image encoder by:

$$g = \frac{\partial s}{\partial A^k},$$

where A^k represents the feature maps and g denotes the gradients. Then we compute the neuron importance weights by average-pooling the gradients:

$$\alpha_k = \frac{1}{Z} \sum_i \sum_j g_{ij}^k.$$

Here, α_k is the neuron importance weights and Z is the number of pixels in each feature map. We then calculate a weighted combination of the feature maps A^k and the neuron importance α_k:

$$L = \text{ReLU}\left(\sum_k \alpha_k A^k\right),$$

an activation function ReLU is applied to filter out all negative activations. Specifically, we use the feature map after the first normalization layer of the last residual block to compute the gradients for CAM.

Integration with CLIP-ES. In summary, the CLIP-ES [7] we adopted is composed of a Grad-CAM and a class-aware attention-based affinity (CAA) module. The CAA module is introduced to enhance the vanilla multi-head self-attention (MHSA) for the Vision Transformer in CLIP. Given an image, class-wise CAM maps $M_c \in R^{h \times w}$ for each target class c and the attention weight $W^{attn} \in R^{hw \times hw}$ are obtained from MHSA. For the attention weight, which is made asymmetric due to the use of different projection layers by the query and key, Sinkhorn normalization [17] is applied (alternately applying row-normalization and column-normalization) to convert it into a doubly stochastic matrix D, and the symmetric affinity matrix A can be derived as follows:

$$A = \frac{D + D^T}{2}, \text{ where } D = \text{Sinkhorn}(W^{attn}). \quad (1)$$

For the CAM map $M_c \in R^{h \times w}$, a mask map for each target class c can be obtained by thresholding the CAM with λ. Then a set of bounding boxes can be generated based on the thresholded masks. These boxes are used to mask the affinity weight matrix A, and then each pixel can be refined based on the masked affinity weight and its semantically similar pixels. This refinement process can be formalized as follows:

$$M_c^{aff} = B_c \odot A^t \cdot \text{vec}(M_c), \quad (2)$$

where $B_c \in R^{1 \times hw}$ represents the box mask obtained from the CAM of class c, $\odot$ denotes the Hadamard product, t indicates the number of refining iterations, and $\text{vec}(\cdot)$ denotes the vectorization of a matrix. It should be noted that the attention map and CAM are extracted in the same forward pass. Therefore, CAA refinement is performed in real time and does not need an additional stage. Our implementation uses the attention maps from the last 8 layers of Vision Transformer for CAA.

3. Implementation Details of Visual Prompts

The Python code of visual prompts is shown in Algorithm A, which is at the end of the supplementary material.

4. Breakdown of Background Tokens

We break down the background tokens into 3 sub-categories for ablation study (experiment results are shown in the main manuscript in Table 6):

- `Terrestrial`: [‘ground’, ‘land’, ‘grass’, ‘tree’, ‘mountain’, ‘rock’, ‘valley’, ‘earth’,‘terrain’, ‘forest’, ‘bush’, ‘hill’, ‘field’, ‘pasture’, ‘meadow’, ‘plateau’, ‘cliff’, ‘canyon’, ‘ridge’, ‘peak’, ‘plain’, ‘prairie’, ‘tundra’, ‘savanna’, ‘steppe’, ‘crag’, ‘knoll’, ‘dune’, ‘glen’, ‘dale’, ‘copse’, ‘thicket’]
- `Aquatic-Atmospheric`: [‘sea’, ‘ocean’, ‘lake’, ‘water’, ‘river’, ‘sky’, ‘cloud’, ‘pond’, ‘stream’, ‘lagoon’, ‘bay’, ‘gulf’, ‘fjord’, ‘estuary’, ‘creek’, ‘brook’, ‘reservoir’, ‘pool’, ‘spring’, ‘marsh’, ‘swamp’, ‘wetland’, ‘glacier’, ‘iceberg’, ‘atmosphere’, ‘stratosphere’, ‘mist’, ‘fog’, ‘rain’, ‘drizzle’, ‘hail’, ‘sleet’, ‘snow’, ‘thunderstorm’, ‘breeze’, ‘wind’, ‘gust’, ‘hurricane’, ‘tornado’, ‘monsoon’, ‘cumulus’, ‘cirrus’, ‘stratus’, ‘nimbus’]
- `Man-Made`: [‘building’, ‘house’, ‘wall’, ‘road’, ‘street’, ‘railway’, ‘railroad’, ‘bridge’, ‘edifice’, ‘structure’, ‘apartment’, ‘condominium’, ‘skyscraper’, ‘highway’, ‘boulevard’, ‘lane’, ‘alley’, ‘byway’, ‘avenue’, ‘expressway’, ‘freeway’, ‘path’, ‘overpass’, ‘underpass’, ‘viaduct’, ‘tunnel’, ‘footbridge’, ‘crosswalk’, ‘culvert’, ‘dam’, ‘archway’, ‘causeway’, ‘plaza’, ‘square’, ‘station’, ‘terminal’]

5. Implementation Details of Mutual Background for Pascal Context

Our approach involves creating a list of background queries to minimize false positive predictions in mask proposals. However, in the Pascal Context dataset [10], many “stuff” categories (*e.g.* sky, ground, sea) serve as background queries for “object” categories (*e.g.* bird, car, boat).

Directly removing these 'stuff' categories from the background query list and generating object and stuff masks using CAM leads to noisy results due to the lack of false positive background suppression. To address this issue, we adopt a mutual background strategy. In this method, object and stuff masks are produced separately, using object categories as the background queries for stuff masks and vice versa. This technique not only maintains the benefit of reducing false positives but also significantly enhances performance in the Pascal Context dataset.

6. Implementation Details of Referring Image Segmentation.

We use ViT-B/16 as the backbone of the visual encoder for both the mask proposal generator and mask classifier, and use `circle` and `background blur` as the visual prompts for the inputs of mask classifier. The η, θ, λ were set to (0.5, 0.3, 0.5), (0.2, 0.1, 0.5), (0.5, 0.1, 0.6) for refCOCO, refCOCO+ and refCOCOg, respectively. All splits of these three datasets share the same set of hyperparameters. We note that we **do not** apply SAM for referring image segmentation.

7. More Visualization Results

7.1. Visualization results on different post-processors

Figures A and B present a comparative visualization of the post-processing techniques Conditional Random Field (CRF) and Segment Anything Model (SAM) [4], applied to randomly chosen samples from the VOC [3] and COCO Object datasets [1]. Initial observations reveal that the application of CRF in CaR facilitates the generation of high-quality masks, albeit with notable limitations in delineating boundaries between distinct semantic masks. The integration of SAM enhances the precision of these masks, yielding clearer and more well-defined boundaries. Nevertheless, the implementation of SAM is not without drawbacks; it occasionally leads to false negative predictions, stemming from mismatches between CaR raw masks and SAM candidate masks (the matching algorithm is introduced in the main manuscript), or false positive predictions due to the overly coarse nature of SAM masks. Meanwhile, we find SAM is not very sensitive to stuff classes, so combining SAM on Pascal Context will not lead to much increase in mIoU.

7.2. Visualization comparison for different open-vocabulary segmentation methods.

Figure C presents a qualitative comparison of open-vocabulary segmentation results for a variety of non-standard subjects, including unique characters, brands, and landmarks. These subjects are notably distinct from common objects. The Grounded SAM [8] method demonstrates proficiency in segmenting prominent objects with precision, yet it often misclassify these segments. The OVSeg [5] approach also generates low-quality segmentation masks and inaccurate class predictions. In contrast, our methodology CaR excels by creating high-quality masks with accurate semantic class predictions, showcasing its superior capability in the realm of open-vocabulary segmentation.

8. Limitation

The primary limitation of our method is that its performance is bounded by the pre-trained VLM. For example, since the CLIP model utilizes horizontal flipping augmentation during training, it becomes challenging for our model to successfully distinguish between the concepts "left" and "right". However, we believe that this issue can be easily resolved through adjustments, such as incorporating better data augmentation techniques during the pre-training phase.

9. Future Potentials and Broader Impact

CaR is simple, straightforward yet highly efficient. To enhance its performance further, we provide two ways to explore. First, incorporating additional trainable modules such as Feature-Pyramid Networks can significantly improve its capability in handling small objects. Second, since our method is fundamentally compatible with various Vision-Language Models (VLMs), it presents an intriguing opportunity to investigate integration with other VLMs. Moreover, CaR can serve the purpose of generating pseudo-labels for other open-vocabulary segmenters.

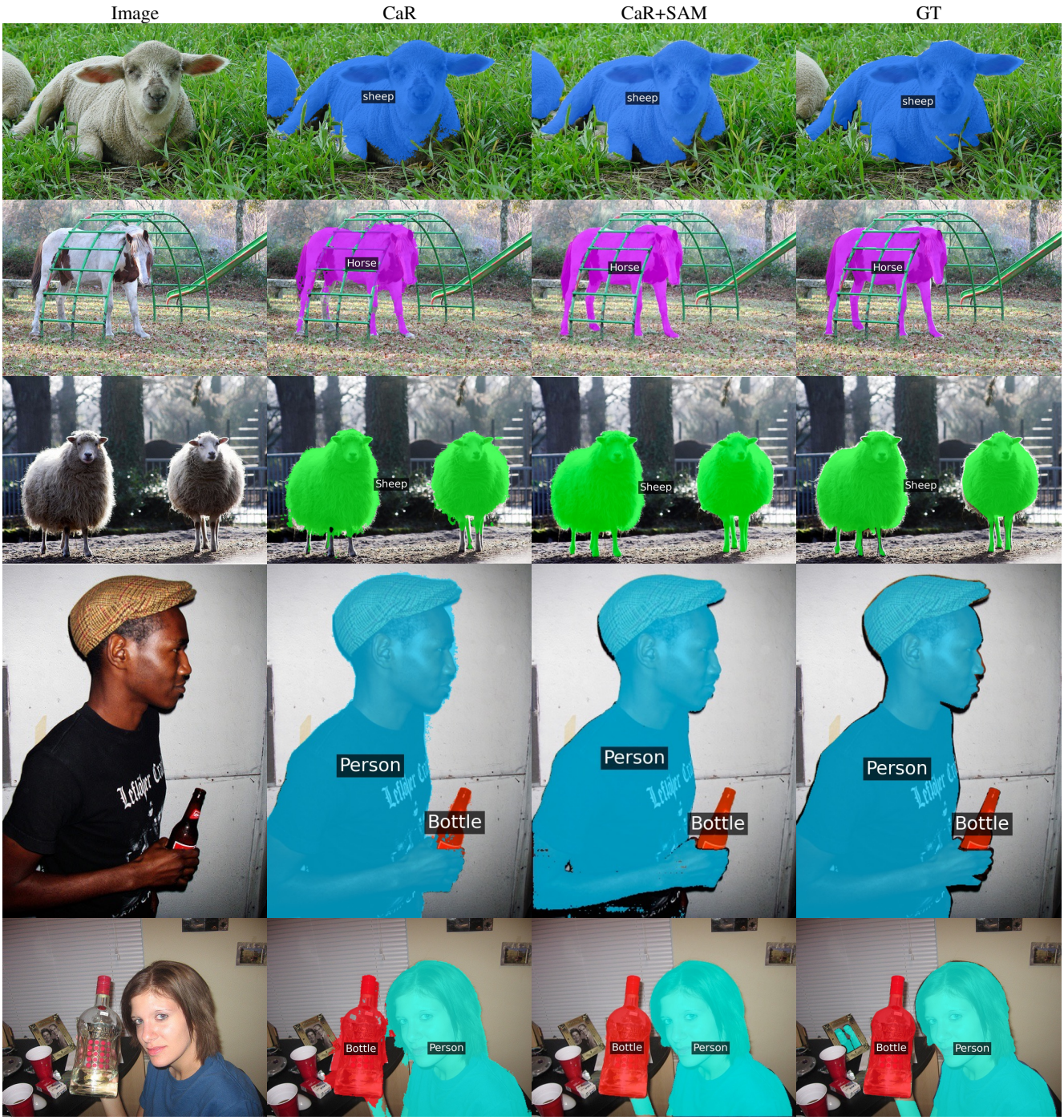

Figure A. Comparison of different post-processors on randomly selected images from PASCAL VOC.

Figure B. Comparison of different post-processors on randomly selected images from COCO Object.

Figure C. Visualization comparison of different open-vocabulary segmentation methods.

Algorithm A Pseudo-code of CLIP as RNN in PyTorch style.

```
import cv2
import numpy as np
import torch
from scipy.ndimage import binary_fill_holes
def apply_visual_prompts(
      image_array,
      mask,
      visual_prompt_type=('circle'),
      visualize=False,
      color=(255, 0, 0),
      thickness=1,
      blur_strength=(15, 15)):
   prompted_image = image_array
   inv_mask = (1 - mask)[:, :, None]
   if 'blur' in visual_prompt_type:
      # blur the part out side the mask
      # Blur the entire image
      blurred = cv2.GaussianBlur(prompted_image,
          blur_strength, 0)
      # Get the sharp region using the mask
      sharp_region = cv2.bitwise_and(
         prompted_image,
         prompted_image,
         mask=np.clip(mask, 0, 255)))
      # Get the blurred region using the inverted mask

      blurred_region = (blurred * inv_mask)
      # Combine the sharp and blurred regions
      prompted_image = cv2.add(sharp_region,
          blurred_region)
   if 'gray' in visual_prompt_type:
      gray = cv2.cvtColor(prompted_image, cv2.
          COLOR_BGR2GRAY)
      # make gray part 3 channel
      gray = np.stack([gray, gray, gray], axis=-1)
      # Get the sharp region using the mask
      color_region = cv2.bitwise_and(
         prompted_image,
         prompted_image,
         mask=np.clip(mask, 0, 255))
      # Get the blurred region using the inverted mask
      inv_mask = 1 - mask
      gray_region = (gray * inv_mask)
      # Combine the sharp and blurred regions
      prompted_image = cv2.add(color_region,
          gray_region)
   if 'black' in visual_prompt_type:
      prompted_image = cv2.bitwise_and(
         prompted_image,
         prompted_image,
         mask=np.clip(mask, 0, 255))
   if 'circle' in visual_prompt_type:
      mask_center, mask_height, mask_width = mask2chw(
          mask)
      center_coordinates = (mask_center[1],
          mask_center[0])
      axes_length = (mask_width // 2, mask_height //
          2)
      prompted_image = cv2.ellipse(prompted_image,
                             center_coordinates,
                             axes_length, 0, 0, 360,
                                 color, thickness)
   if 'rectangle' in visual_prompt_type:
      mask_center, mask_height, mask_width = mask2chw(
          mask)
      center_coordinates = (mask_center[1],
          mask_center[0])
      start_point = (mask_center[1] - mask_width //
                2, mask_center[0] - mask_height // 2)
      end_point = (mask_center[1] + mask_width //
               2, mask_center[0] + mask_height // 2)
      prompted_image = cv2.rectangle(prompted_image,
                               start_point,
                               end_point,
                               color, thickness)
   if 'contour' in visual_prompt_type:
      # Find the contours of the mask
      # fill holes for the mask
      mask = binary_fill_holes(mask)
      contours, hierarchy = cv2.findContours(mask, cv2
          .RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
      # Draw the contours on the image
      prompted_image = cv2.drawContours(
         prompted_image, contours, -1, color,
             thickness)
   return prompted_image
```

References

[1] 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. 3

[2] 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

[3] 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. 1, 3

[4] 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. 3

[5] 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, 3, 6

[6] 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. 1

[7] 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. 1, 2

[8] 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. 3, 6

[9] 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. 1

[10] 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. 1, 2

[11] 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. 1

[12] 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. 1

[13] 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

[14] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *IJCV*, 115:211–252, 2015. 1

[15] 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. 1

[16] 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

[17] Richard Sinkhorn. A relationship between arbitrary positive matrices and doubly stochastic matrices. *The annals of mathematical statistics*, 35(2):876–879, 1964. 2

[18] 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

[19] 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. 1

[20] 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