Exploring Simple Open-Vocabulary Semantic Segmentation

Zihang Lai
Visual Geometry Group (VGG), University of Oxford
zlai@robots.ox.ac.uk

Abstract

Open-vocabulary semantic segmentation models aim to accurately assign a semantic label to each pixel in an image from a set of arbitrary open-vocabulary texts. In order to learn such pixel-level alignment, current approaches typically rely on a combination of (i) image-level VL model (e.g. *CLIP), (ii) ground truth masks, (iii) custom grouping encoders, and (iv) the Segment Anything Model (SAM). In this paper, we introduce S-Seg, a simple model that can achieve surprisingly strong performance* ***without depending on any of the above elements****. S-Seg leverages pseudo-masks and language features to train a MaskFormer, and can be easily trained from publicly available image-text datasets. Contrary to prior works, our model directly trains for pixel-level features and language alignment. Once trained, S-Seg generalizes well to multiple testing datasets without requiring fine-tuning. In addition, S-Seg has the extra benefits of scalability with data and consistently improving when augmented with self-training. We believe that our simple yet effective approach will serve as a solid baseline for future research. Project page:* `zlai0.github.io/S-Seg`.

1. Introduction

Open-vocabulary semantic segmentation presents a unique challenge as it requires assigning accurate semantic labels to each pixel in an image using arbitrary open-vocabulary texts, rather than a fixed set of classes. This means that the model must be able to segment and classify any arbitrary categories expressed in language. Achieving this requires a robust, pixel-level alignment between images and textual descriptions, which enables accurate association of each pixel with the most relevant class from a dynamically provided set of textual categories.

A primary obstacle in this domain is that it is impossible to construct datasets that provide pixel-level annotations for *all* possible labels. This limitation often results in the adoption of weakly-supervised or semi-supervised learning approaches. Current methods typically rely on a combination of strategies to learn the required pixel-level alignment.

Figure 1. **S-Seg result on a web image.** Our goal is to segment everything, including fictional characters like *minions*.

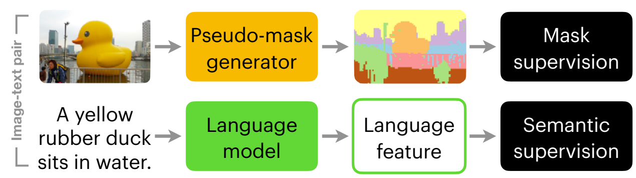

Figure 2. Our **S-Seg** framework leverages pseudo-mask and language to train a MaskFormer. We show that our method of directly training for pixel-level feature and language alignment yields superior results.

One common tactic is adapting existing Vision-Language (VL) models, which are initially trained for image-level alignment (*e.g.*, CLIP [37]), to perform at the pixel level. Another strategy involves training models on ground truth masks that are annotated for a select number of *seen* classes, thereby encouraging the model to extrapolate its knowledge to novel *unseen* classes during testing. Furthermore, specialized models such as GroupViT [47] and OVSegmentor [48], which are explicitly designed for open-vocabulary segmentation, are being explored. Lastly, recent literature adapts the Segment Anything Model (SAM), originally not designed to classify segments, for effective use in open-vocabulary segmentation.

In this paper, we report a model that can work surprisingly well with *none* of the above strategies. Our approach, named S-Seg, is built on top of a standard MaskFormer model. Our model directly trains for pixel-level feature and language alignment, using *neither* existing large image-level alignment models like CLIP [37] *nor* manually annotated segmentation or classification labels.

One of the biggest challenges we face is finding the right

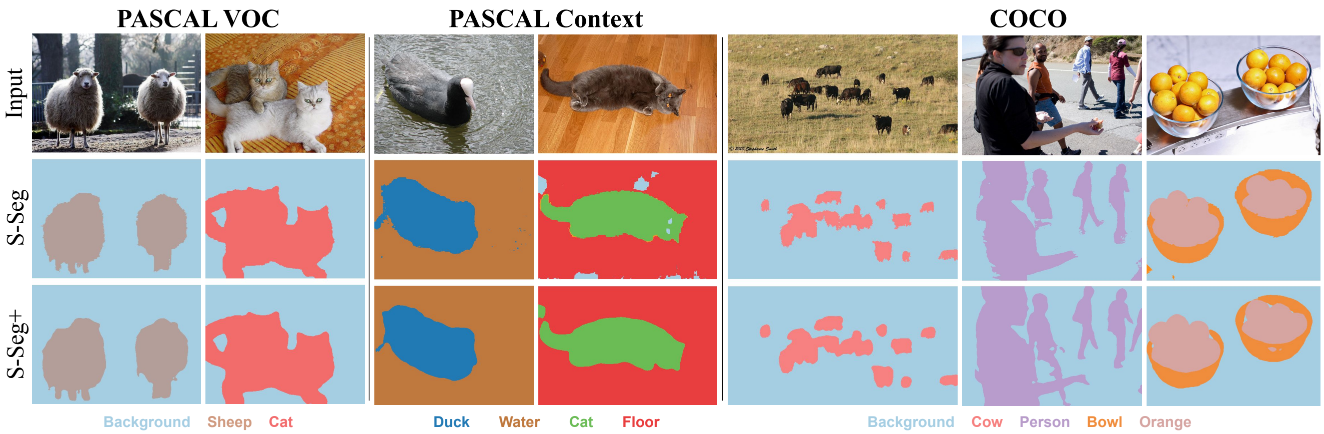

Figure 3. **Qualitative results of S-Seg, evaluated using *all* dataset classes as queries.** Our model copes with challenging situation, such as overlapping objects (col. 2) and small objects (col. 5). Our model is also capable of handling "stuff" categories such as water and floor (col. 3, 4). Moreover, our S-Seg+ model is able to correct small errors observed in the S-Seg method (col. 4). Finally, in the COCO dataset, which featured a significantly higher number of objects, our model is still able to achieve high accuracy in its predictions.

supervision since annotated masks and labels are not available. To address this issue, we propose to leverage *pseudo-masks* and *language* to supervise MaskFormer. Our strategy involves using a pseudo-mask generator to provide class-agnostic mask supervision by generating pseudo ground truth masks. We adopt a simple design that clusters image representations obtained through self-supervised representation learning methods like DINO [3]. Our experiments demonstrate that this approach delivers exceptional performance, which is essential for high-quality supervision, as well as rapid processing speed, which is necessary for efficient training. In addition, we use noisy web texts to provide semantic supervision. The image-text dataset contains a wide range of concepts and has demonstrated impressive zero-shot classification results [37]. We utilize a straightforward image-text contrastive loss, which has proven to be highly effective. Once trained, our model generalizes well to new categories without requiring fine-tuning.

S-Seg is a simple and effective model that can be trained using publicly available image-text datasets, such as Conceptual Captions [5, 40]. This makes it easy to reproduce and extend for further research. The S-Seg framework is also designed to be flexible with easily replaceable submodules. We prioritize *simplicity* in our subcomponent selection to focus on the general design of our framework, while remaining open to more advanced techniques that could result in further improvements.

We conducted a thorough evaluation of S-Seg using multiple benchmark datasets, and we show that our method achieve competitive results on three widely tested benchmarks (Pascal VOC, Pascal Context, and COCO). In addition, pseudo-mask and language provide scalable supervision and our model consistently improves in performance as more data became available. Finally, we find adding an additional self-training step leads to an even greater improvement to our model, with an average increase of 5.5% mIoU over three datasets, highlighting the potential for further improvement of our approach.

Our simple solution suggests that the reliance on complex models and extensive ground truth data in open-vocabulary semantic segmentation may be reduced, leading to more streamlined and accessible framework for future developments in the field, and we hope our exploration can serve as a solid baseline for future research.

2. Related work

Open-vocabulary segmentation. The earliest efforts to employ language for image segmentation can be traced back to Duygulu et al.'s seminal work [13], where the authors tackled image segmentation by framing it as a machine translation problem. Current approaches leverage a combination of strategies to learn pixel-level image-text alignment.

Adapting image-level vision-language models. The first strategy involves the adapting pretrained vision-language models, originally designed for image-level alignment, to the more granular task of pixel-level alignment. This strategy is widely adopted in open-vocabulary methods [4, 15, 18, 19, 22, 24, 27, 32, 33, 38, 44, 48, 49, 52]. These works vary in their methods of refining image-level models for finer alignment tasks.

MaskCLIP [52] demonstrates modifying the CLIP image encoder can significantly enhance its pixel-level alignment capabilities without requiring retraining. TCL [4] employs CLIP for initial text-to-image region grounding, followed by contrastive learning to refine the alignment between the text embedding and the grounded region. OpenSeg [15] fine-tunes ALIGN [20] using a grounding

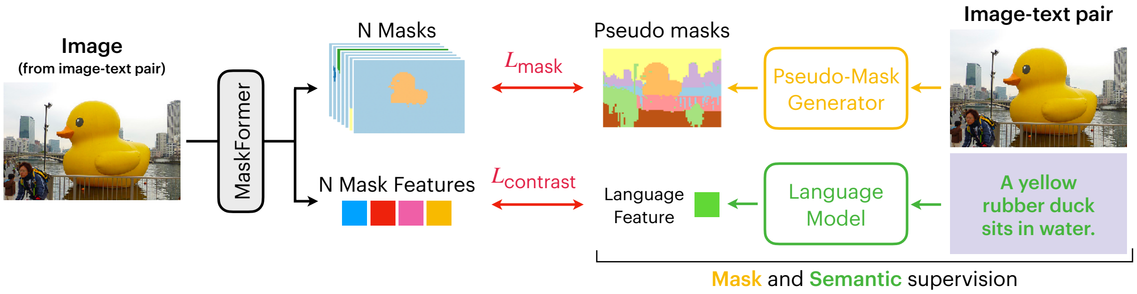

Figure 4. **Overview of S-Seg.** A MaskFormer model computes masks and mask features from an image input. A pseudo-mask generator produces segmentation maps to supervise mask predictions, while a text that describes the image, encoded by a language model trained together with the MaskFormer, provides supervision for mask features using image-text contrastive loss.

loss [17] to better align words in captions to segmentation masks. OpenSeg and DiffuMask [44] also explored the use of pseudo-masks. The primary distinction lies in their dependency on different sources for learning; OpenSeg uses annotated segments while DiffuMask employs masks generated through diffusion. In contrast, our method is entirely learned from pseudo masks. Also, our mask generator is entirely self-supervised, whereas their mask generator is fully-supervised.

Ground truth masks. Another effective strategy [11, 15, 18, 19, 24, 27, 33] involves training models using ground truth masks annotated for a limited set of seen classes. By training on seen annotations, models are encouraged to learn detailed features and patterns that are potentially applicable beyond the scope of the trained classes.

Of most relevance, ZegFormer [11] trains a MaskFormer by decoupling zero-shot semantic segmentation into two sub-tasks, a class-agnostic grouping task and a zero-shot segment classification task. Our method has similar training paradigm but with notable distinctions. Similar to GroupViT, we train exclusively with image-text pairs and do not utilize a pretrained CLIP model. Notably, even without access to ground truth masks, labels, or CLIP, our method outperforms ZegFormer in unseen categories, indicating potentially stronger generalization.

Custom grouping-based encoders. The third strategy employs custom-designed models specifically for open-vocabulary segmentation. GroupViT [47] groups pixels in an image hierarchically based on their attention scores with learnable group tokens. OVSegmentor [48] applies Slot Attention [31] for a similar pixel grouping process based on feature proximity.

Segment Anything Model (SAM). The final strategy adapts the Segment Anything Model (SAM) [23], a powerful segmentation model trained on over 11 million images and 1.1 billion masks, for open-vocabulary segmentation. While the SAM model segments object with high accuracy, it does not effectively classify these segments. To address this limitation, follow-up works has adapted the SAM model for grounding, closed-set, and open-vocabulary segmentation [25, 39, 51] by leveraging additional training

```
# I [n, h, w, c] - minibatch of aligned images
# T [n, l]       - minibatch of aligned texts
# N              - number of MaskFormer queries
# C              - number of pseudo masks

# predict mask, mask feature, and text feature
M, M_f = maskformer(I) # [n, N, H, W], [n, N, d_f]
T_f = text_encoder(T)  # [n, d_f]

# aggregate all mask features [n, d_f]
M_f = M_f.mean(axis=1)

# generate pseudo masks [n, C, H, W]
S = pseudo_mask_generator(I)

# compute loss
loss_c = contrastive_loss(M_f, T_f)
loss_m = mask_loss(M, S)
loss = (loss_c + loss_m)/2
```

Figure 5. **Pseudocode for training S-Seg with image-text pairs.**

data, adaptors, and distillation.

Our model, S-Seg, can be conceptualized as a synergy of these approaches. It can be viewed as a CLIP model integrated with a MaskFormer image encoder, directly optimizing for pixel-level feature and language alignment. Alternatively, it resembles "ZegFormer with pseudomask and language training" or "GroupVit with a MaskFormer as the grouping mechanism." Interestingly, our model relates to each method by omitting certain core architectural components or supervision method. Even so, our method is able to achieve competitive performance.

3. Approach

Our proposed method, called *S-Seg*, is conceptually simple: we learn a MaskFormer model from *pseudo-mask* and *language*. Our method leverages image-text pairs solely, without relying on ground truth masks or large-scale preatrained models. In figure 5, we provide pseudocode for the core implementation of training S-Seg. Figure 4 provides a schematic layout of our approach.

3.1. Problem definition

We consider the problem of open-vocabulary semantic segmentation, where we aim to learn a function f that maps an image I and a set of category names $\mathcal{C} = \{c_i\}_{i=i}^{N_\mathcal{C}}$ to a semantic segmentation map S, where c_i can be any category name expressed as open vocabulary texts.

Our approach adopts the **unsupervised setting** [4, 38, 47, 52], which aims to learn open-vocabulary segmentation from image-text pairs only, without learning from any dense annotation or class labels. Specifically, we use a web dataset of image-text pairs $\mathcal{D} = \{(I_i, T_i)\}_{i=i}^{N_\mathcal{D}}$ for training, where T_i is a caption that describes the corresponding image I_i. However, since the textual labels are gathered from the web, they may be noisy and contain errors. We **do not** use any additional manually annotated segmentation or classification labels during training.

During testing, a set of category names $\mathcal{C}$ is provided, and the model is tasked with assigning a semantic label $c_i \in \mathcal{C}$ to each pixel in an unlabeled image. The performance of the model is evaluated based on its mean Intersection over Union (mIoU) with the ground truth labels.

3.2. Adapting MaskFormer

Our approach builds on top of MaskFormer [8]. Here, we begin by briefly review MaskFormer and explain the adjustments we made.

The Maskformer model takes an image as input and generates N masks and mask features. First, the input image passes through a backbone model to produce feature maps at different output resolutions. These image features are then fed into a per-pixel encoder, which upsamples and aggregates them into a set of feature maps with higher resolution. Meanwhile, a transformer decoder uses N learnable queries to cross-attend to the set of features with the lowest resolution and gather global information about each segment.

In the original Maskformer, a linear classifier and softmax activation were applied to the output of the decoder to predict class probabilities for a fixed list of categories. However, as we do not have a fixed list of categories, we remove this classifier branch and output the N raw mask features instead.

In addition to predicting mask features, the Maskformer also predicts N binary masks. To predict each mask, a dot product is taken between the mask embedding, generated from mask features, and the high resolution per-pixel feature. Finally, N mask-feature pairs are combined to generate the output.

3.3. S-Seg

S-Seg employs MaskFormer as its segmentation model, but in our weakly-supervised learning setting (where only texts are available), we face the challenge of not having annotated masks and labels. To overcome this, we utilize pseudo labels and language to as supervision.

Our training framework is illustrated in Figure 4. We first generate a set of segmentation maps using our pseudo-mask generator (Sec. 3.4) and use them as supervision for mask prediction. Meanwhile, we use a language model to process input text and generate language embeddings. These embeddings provide supervision for mask features by leveraging image-text contrastive loss (Sec. 3.5).

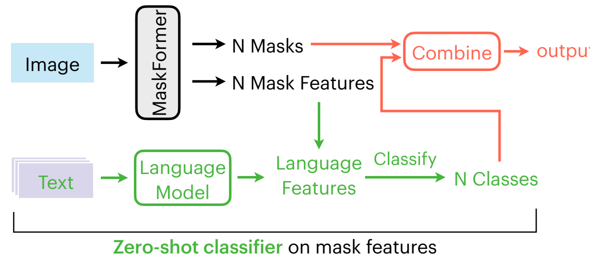

Figure 6. **Testing on S-Seg.** During inference, S-Seg generalize to new categories by leveraging language features generated from a list of candidate classes in text.

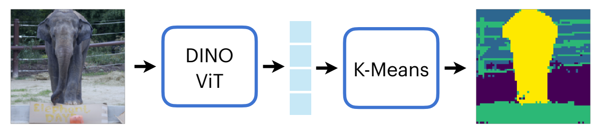

Figure 7. **Pseudo-mask generator** generates pseudo-masks to supervise predicted mask during training. This module takes an image as its input, extracts its features using a DINO pre-trained ViT, and then employs K-means clustering to group the pixels into segments.

Notably, unlike the supervised learning setting, where mask and label annotations are coupled, we *decouple* mask and semantic supervision. This enables us to utilize pseudo-mask and language as two distinct forms of supervision.

In the testing phase (as shown in figure 6), the trained MaskFormer model predicts N masks and mask features from the input image. The language model takes as input a list of candidate category names (represented as texts) and extracts a set of language features. These features are then used to classify the mask features. This process is similar to the one used in CLIP [37], where the image and possible text inputs are encoded by their respective encoders to compute feature embeddings. The cosine similarity between these embeddings is calculated and adjusted by a learnable temperature parameter. The resulting values are normalized into a class probability distribution using a softmax function, and a combination module is used to takes N mask-class pairs to produce the final segmentation map, similar to [8].

Next, we provide a details of the subcomponents in our framework.

3.4. Pseudo-Mask Generator

In our approach, we use a pseudo-mask generator (fig. 7) to produce a class-agnostic segmentation map from the input

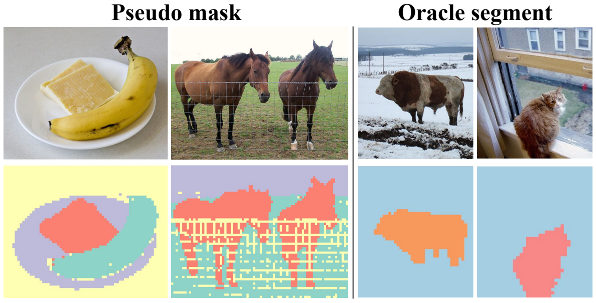

Figure 8. **Example pseudo-masks.** Our pseudo-mask generator is capable of generating high-quality artificial masks. When provided with an oracle label, these masks demonstrate a high degree of overlap with the ground truth annotations.

Method	Sup.	P. VOC↑	P. Context↑	Time(s)↓
Spectral Clus. [41]*	none	49.2	43.2	0.543
K-Means [21]*	none	49.5	43.3	0.188
ImageNet [12]+[21]	label	68.8	58.1	0.079
GroupViT [47]	text	73.7	54.6	**0.002**
Pseudo-mask (Ours)	self	**78.8**	**66.3**	**0.002**

Table 1. **Our pseudo-mask generator achieves excellent oracle performance with rapid speed**, making it an ideal mask supervision. We report amortised running time on a batch of 128 samples, simulating training time scenario. * We process downsampled image at $\frac{H}{8} \times \frac{W}{8}$ resolution to obtain reasonable running time.

image, which supervises the mask prediction of our model.

To implement the pseudo-mask generator, we adopt a simple strategy that involves clustering tokens extracted from a self-supervised pre-trained ViT. Specifically, we use a DINO-pretrained ViT to compute a set of featurized tokens from the input image. We then apply a clustering algorithm (K-Means in our case) to these tokens, assigning each token a label that corresponds to the index of the cluster it belongs to. We reshape the resulting label map into an image and resize it to the original resolution to supervise the mask prediction of our segmentation model.

Despite its simplicity, our pseudo-mask generator achieves both impressive performance, which is crucial for high-quality supervision, and fast processing speed, which is essential for efficient training. We evaluate its performance and compare against baseline methods, and the quantitative results are presented in Table 1, with example predictions visualized in 8. Our method significantly outperforms simple baselines such as K-Means and Spectral Clustering, which naively cluster image pixels, while running two orders of magnitude faster. We also observed that clustering DINO representation outperforms clustering ImageNet pre-trained ViT representation by a significant margin. Notably, our pseudo-mask generator even outperforms GroupViT, which has already employed vision-language training.

Since the predicted masks are unordered, we need to match the N predicted masks with K pseudo ground truth masks. To accomplish this, we utilize bipartite matching, as described in [2, 8], which assigns a pseudo-mask to each predicted mask such that the overall assignment cost is minimal in all possible assignments. Since each pseudo-mask is assigned to at most one predicted mask, $N-K$ pseudo-masks are unassigned to no-object. Unlike standard training recipe [8], we do not penalize these no-object masks, nor do we use classification loss as an assignment cost. Finally, we compute the mask loss between predicted masks and their corresponding pseudo-mask, utilizing a combination of dice loss [34] and focal loss [29].

$$\mathcal{L}_{\text{mask}} = \lambda_{\text{dice}}\mathcal{L}_{\text{dice}} + \lambda_{\text{focal}}\mathcal{L}_{\text{focal}} \tag{1}$$

3.5. Language Supervision

Our model learns to classify open-vocabulary concepts from language supervision. To train the model, we use an image-text contrastive loss [15, 37]. Specifically, we view N mask features as representation of the input image, each capturing information about a different part of the image. We then compute a single feature that represents the entire image by taking the average of these mask features. To encode the text, we use a text transformer [42] and select the embedding corresponding to the `[EOS]` token, resulting in a textual feature. Since the visual and textual features may have different dimensions, we project each representation into a common embedding space using 2-layer MLPs. To compute the image-text contrastive loss, we calculate the cosine similarity between the image embeddings and the text embeddings within the same batch. Following common practice [26, 36, 37], we decouple the image-text contrastive loss into two parts:

$$\mathcal{L}_{I\rightarrow T} = -\frac{1}{N}\sum_{i}^{N}\log\frac{\exp(x_i^{\intercal}y_i/\sigma)}{\sum_{j=1}^{N}\exp(x_i^{\intercal}y_j/\sigma)} \tag{2}$$

$$\mathcal{L}_{T\rightarrow I} = -\frac{1}{N}\sum_{i}^{N}\log\frac{\exp(y_i^{\intercal}x_i/\sigma)}{\sum_{j=1}^{N}\exp(y_i^{\intercal}x_j/\sigma)} \tag{3}$$

where x_i and y_i are L2-normalized embedding of image and text of the i-th pair. N denotes batch size and σ is a learnable temperature parameter optimized together with the rest of the model. The total loss is the sum of these two losses, $\mathcal{L}_{\text{contrastive}} = \mathcal{L}_{I\rightarrow T} + \mathcal{L}_{T\rightarrow I}$.

3.6. Training Loss

Overall, mask loss (Sec. 3.4) and image-text contrastive loss (Sec. 3.5) complete the necessary mask and semantic supervision that is needed to train our model. The final loss is a weighted combination of the two losses:

$$L = \lambda_{\text{mask}}\mathcal{L}_{\text{mask}} + \lambda_{\text{contrastive}}\mathcal{L}_{\text{contrastive}} \tag{4}$$

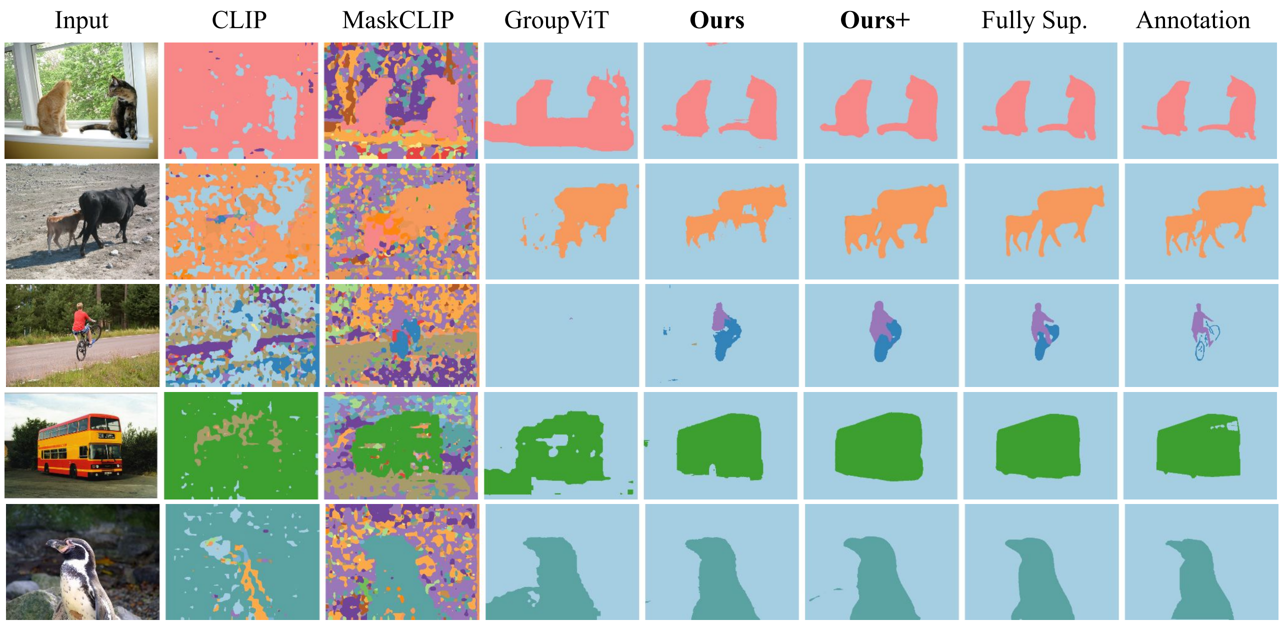

Figure 9. **Qualitative comparison with existing methods.** CLIP [37] is primarily designed for classification and does not perform well in segmentation. MaskCLIP [52] adapts CLIP for segmentation, although it produces noisy predictions and cannot handle background classes. GroupViT [47] is a strong competitor, but it could struggle in challenging scenarios.

In our experiment, we use $\lambda_{\text{mask}} = 1.0$, $\lambda_{\text{contrastive}} = 1.0$, $\lambda_{\text{dice}} = 1.0$, $\lambda_{\text{focal}} = 20.0$.

3.7. Self-training

To further improve our results, we introduce an optional step wherein we train a new model using the predictions generated by our current model. This process of self-training results in an augmented model, which we refer to as S-Seg+. More specifically, when we evaluate on a given dataset, we generate pseudo labels for the unlabeled images in the training set. Subsequently, we employ these pseudo labels to train a new segmentation model.

Self-training improves the accuracy by leveraging additional data [46], augmentation [53], and bootstrapping [16]. In our situation, self-training offers even greater benefits since we can take advantage of additional information that is obtainable during testing: unlabeled images and testing categories. We show that this additional step improves our results significantly at no extra manual labelling cost.

4. Experiments

In this section, we empirically evaluate our method and compare to existing approaches. We show that, although our method is quite simple, it performs surprisingly well against more complex existing methods.

4.1. Implementation details

Architecture. Our experiments use MaskFormer [8] with a 6-layer transformer decoder and $N = 64$ queries. The hidden and output feature dimension is 256. The language model is a Transformer [42] with 12 layers, each with a hidden dimension of 256. We use a 2-layer MLP to project the visual and text feature into a common embedding space. We use DINO ViT-S/8 as the pretrained ViT in pseudo-mask generator.

Training. During training, we used three publically available datasets: CC3M [40], CC12M [5], and RedCaps [10], containing 3M, 12M and 12M image-text pairs, respectively. Due to storage constraint, we use only first 11M data samples at a smaller resolution of 448×448 when using RedCaps dataset. In total, we use at most 26M image-text pairs for training - this is an order of magnitude fewer data than CLIP [37] and 1-4M fewer than GroupViT [47]. For more detailed training hyperparameters, please refer to the appendix.

Inference. We evaluate S-Seg on the validation set of three datasets: Pascal VOC 2012 [14], Pascal Context [35] and COCO [28]. The Pascal VOC dataset contain 1449 images for testing. Each image is labeled with 20 foreground classes and a background class. The Pascal Context dataset contains 5104 testing images with 59 foreground classes and a background class. The COCO dataset contains 5000 images for testing with 80 foreground classes and an additional background class. Following GroupViT [47], we threshold the maximum probability to obtain background prediction. During inference, we set the input resolution to 448×448, in consistent with [47].

Method	OV	Mask	VLM	Enc.	SAM	P. VOC	P. Context	COCO
Linearly-probed classification models:								
MoCo v3 [7]	✗	-	-	-	-	34.3	21.3	-
DINO [3]	✗	-	-	-	-	39.1	20.4	-
Open-vocabulary models:								
CLIP [37]†	✓	✗	✓	✗	✗	13.5	8.1	5.9
MaskCLIP [52]†	✓	✗	✓	✗	✗	26.8	22.8	12.8
ViL-Seg [30]	✓	✗	✓	✗	✗	34.4	16.3	16.4
CLIP_{py} [38]	✓	✗	✓	✗	✗	52.2	-	-
GroupViT [47]	✓	✗	✗	✓	✗	50.8	23.7	27.5
SegCLIP [32]	✓	✗	✓	✗	✗	52.6	24.7	26.5
OVSegmentor [48]	✓	✗	✗	✓	✗	53.8	20.4	25.1
TCL [4]	✓	✗	✓	✓	✗	55.0	**30.4**	**31.6**
SAM-CLIP [43]	✓	✓	✓	✗	✓	**60.6**	29.2	31.5
S-Seg (Ours)	✓	✗	✗	✗	✗	53.2	27.9	30.3
S-Seg+ (Ours)	✓	✗	✗	✗	✗	**62.0**	**30.2**	**35.7**
Fully-supervised segmentation models:								
DeepLabV3+† [6]	✗	-	-	-	-	78.7	46.4	55.7
MaskFormer† [8]	✗	-	-	-	-	81.2	50.0	62.1

Table 2. **Open-vocabulary (OV) semantic segmentation results (background pixels included in evaluation).** Benchmarked following standard evaluation protocols for *unsupervised* open-vocabulary models trained without annotated masks [4, 32, 47, 48]. Models labels: *Mask* (use GT mask), *VLM* (employ pre-trained large VL model like CLIP), *Enc.* (require specialized grouping encoder), and *SAM* (use the SAM model). Our approach obtain second highest performance on average, without using any extra components such as CLIP and SAM. † denotes our recomputed results. Higher values are better.

Method	P. VOC	P. Context	COCO	3-Avg.
B1: Pseudo Mask + CLIP	12.9	3.9	2.9	6.6
B2: Pseudo-mask ViT	23.2	11.0	10.4	14.9
S-Seg (Ours)	**44.9**	**22.9**	**22.5**	**30.1**

Table 3. **Simple baselines for open-vocabulary semantic segmentation.** All models are trained on CC12M. Higher values are better. Two simple baselines fail to obtain satisfactory results, even using after using our pseudo masks and no less training data.

4.2. Simple baselines

The high quality of pseudo-masks (as shown in Figure 7) may lead one to assume that the primary challenge is simply classifying these masks, and that this can be accomplished by utilizing pre-existing methods such as CLIP. To test this assumption, we first develop two simple baselines.

Baseline 1: Pseudo-mask + CLIP. Firstly, our pseudo label generator is utilized to obtain pseudo segments. Then, we iterate through all the masks and apply the current mask to the original image. Next, the masked image is fed to CLIP for classification and the resulting class label is assigned to the corresponding segment.

Baseline 2: Pseudo-mask ViT. We introduce a new visual backbone that differs from the regular ViT. Instead of pooling all image tokens into a single feature, we first individually pool tokens in each segment of the pseudo-mask into segment features, and then pool these features into a visual embedding. We train a CLIP-like model from scratch using this visual backbone. During testing, we classify each segment feature and assign the label to that segment.

Method	OV	Mask	VLM	Enc.	SAM	P. VOC	P. Context	COCO
Open-vocabulary models (annotated masks required for training):								
ZS3Net [1]		✓	✗	✗	✗	38.3	19.4	21.1
LSeg [24]		✓	✓	✗	✗	52.3	-	27.2
OpenSeg [15]		✓	✓	✗	✗	77.2	45.9	38.1
ZegFormer [11]		✓	✓	✗	✗	80.7	-	-
GKC [19]		✓	✓	✗	✗	83.2	45.2	-
ODISE [49]		✓	✓	✓	✗	84.6	57.3	65.2*
DeOp [18]		✓	✓	✗	✗	91.7	48.8	-
OVSeg [27]		✓	✓	✗	✗	94.5	55.7	-
SAN [33]		✓	✓	✓	✗	94.6	57.7	-
FC-CLIP [50]		✓	✓	✓	✗	95.4	58.4	-
SED [45]		✓	✓	✓	✗	96.1	60.6	-
CAT-Seg [9]		✓	✓	✓	✗	**96.6**	**62.0**	-
Open-vocabulary models (annotated masks not required for training):								
CLIP [37]†	✓	✗	✓	✗	✗	39.6	9.0	13.8
MaskCLIP [52]†	✓	✗	✓	✗	✗	49.5	25.5	23.6
GroupViT [47]†	✓	✗	✗	✓	✗	77.2	23.0	37.5
TCL [4]	✓	✗	✓	✓	✗	**83.2**	**33.9**	-
S-Seg (Ours)	✓	✗	✗	✗	✗	81.8	27.2	42.4
S-Seg+ (Ours)	✓	✗	✗	✗	✗	**84.7**	**31.6**	**53.0**
Fully-supervised segmentation models:								
DeepLabV3+† [6]	✗	GT	-	-	-	89.9	48.5	66.9

Table 4. **Open-vocabulary (OV) semantic segmentation results (background pixels excluded in evaluation).** Benchmarked following standard protocol for evaluating *supervised* open-vocabulary models trained with annotated masks [11, 18, 19, 27, 33]. Similar to the previous setting, S-Seg achieves competitive performance compared to earlier methods trained without using any extra components such as CLIP and SAM. *COCO is used for training. Higher values are better.

The results are presented in Table 3. As we can see, open-vocabulary segmentation is more complex than simply grouping image into segments and then categorizing them into classes, even when the segments are of high quality. Baseline 1 employs a significantly larger pretrained CLIP ViT/L-14 model that was also trained on a much larger dataset, while Baseline 2 is trained using the same data as ours. Nevertheless, both baselines fail to achieve satisfactory results, suggesting that open-vocabulary segmentation cannot be naively deconstructed in such ways. We hypothesize that a multi-task learning approach that jointly trains for segmentation and classification could yield significant advantages.

4.3. Evaluation with background

In table 2, we evaluate our model and compare with existing method on open-vocabulary semantic segmentation task. Following standard evaluation protocols on open-vocabulary model trained *without* annotated masks [4, 32, 47, 48], we include background pixels in evaluation and obtain background prediction by setting a theshold for back-

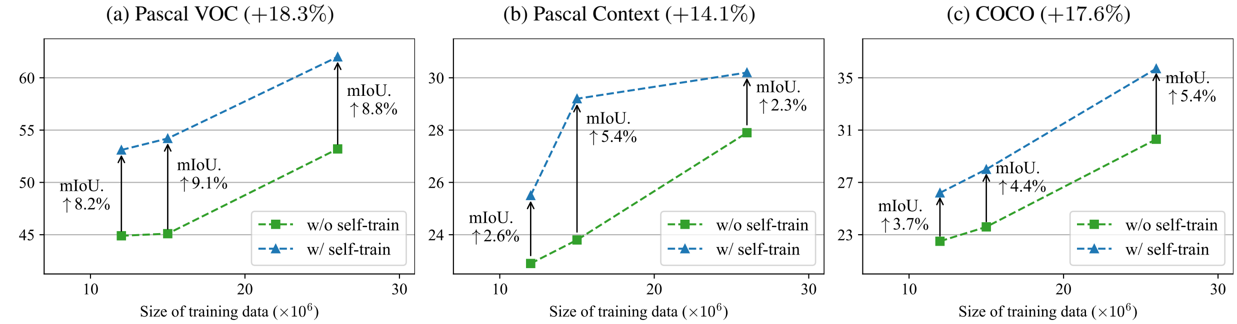

Figure 10. **Self-training improvement.** We show average relative improvement in bracket on top of the plot. we observe that self-training consistently leads to significant improvement for S-Seg across all of our training and testing data settings.

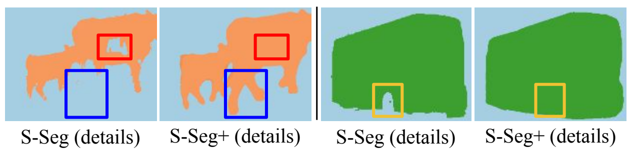

Figure 11. **Visualizing effect of self-training.** Our self-trained S-Seg+ model demonstrates the ability to accurately predict in regions overlooked by S-Seg, as shown in the colorful rectangles.

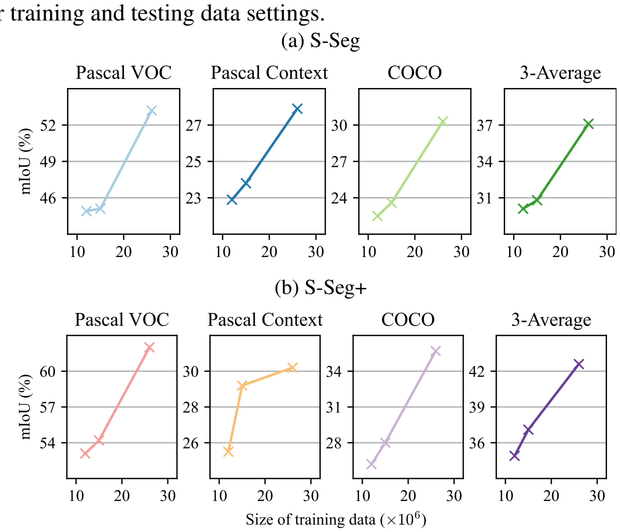

Figure 12. **Scaling training data provide consistent gain in performance, with or without self-training.** We train our model using different sizes of data: CC12M (12M), CC12M+CC3M (15M), and CC12M+CC3M+RedCaps (26M). We note a steady improvement in the performance as the data size increases.

ground classes [47]. Despite the simplicity of S-Seg, our approach achieve competitive performance over previous open-vocabulary segmentation methods that does not require mask annotations. Our model has second highest performance on average and has better results than GroupViT on all datasets. Moreover, our self-trained model, S-Seg+, provides an impressive 5.5% mIoU improvement over our base model S-Seg (42.6% vs 37.1% 3-avg. mIoU), suggesting the efficacy of self-training.

4.4. Evaluation without background

We also evaluate our model on the evaluation protocol commonly used for evaluating open-vocabulary models *with* annotated masks [11, 18, 19], where the background pixels are excluded in evaluation. We note that this setting is easier because background class is more diverse in appearance and often requires additional processing such as thresholding. Table 4 shows the results. Similar to the previous setting, our S-Seg and S-Seg+ models achieve competitive performance compared to earlier methods.

4.5. Ablation studies

Self-training. We investigated the effectiveness of self-training for improving segmentation performance. To this end, we compared S-Seg and S-Seg+ on three datasets and evaluated the results using Figure 10. We found that self-training consistently improved the segmentation performance by a significant margin (+5.5% mIoU on average), regardless of the data size and test dataset. These results indicate that self-training is a reliable approach for enhancing the performance of S-Seg and can provide a desirable complement for further improvement.

Data scalability. To evaluate the scalability of our method, we trained S-Seg and S-Seg+ using three datasets of increasing sizes: 12M, 15M, and 26M. The results of the experiments are presented in Figure 12. We observed that both models achieve significant improvements in performance across all three testing datasets as the amount of data increased, suggesting that our method scales well with larger datasets.

5. Conclusion

To summarize, we propose S-Seg, a simple and intuitive framework that enables accurate and generalizable open-vocabulary segmentation. Our algorithm directly trains for pixel-level feature and language alignment, and does not require manual segmentation annotations or extensive pretraining. We hope that our simple yet effective approach will serve as a solid baseline for future research.

References

[1] Maxime Bucher, Tuan-Hung Vu, Matthieu Cord, and Patrick Pérez. Zero-shot semantic segmentation. *NeurIPS*, 2019. 7

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020. 5

[3] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021. 2, 7

[4] Junbum Cha, Jonghwan Mun, and Byungseok Roh. Learning to generate text-grounded mask for open-world semantic segmentation from only image-text pairs. In *CVPR*, 2023. 2, 4, 7

[5] Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *CVPR*, 2021. 2, 6

[6] 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. 7

[7] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *ICCV*, 2021. 7

[8] Bowen Cheng, Alex Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. 2021. 4, 5, 6, 7

[9] Seokju Cho, Heeseong Shin, Sunghwan Hong, Anurag Arnab, Paul Hongsuck Seo, and Seungryong Kim. Cat-seg: Cost aggregation for open-vocabulary semantic segmentation. In *CVPR*, 2024. 7

[10] Karan Desai, Gaurav Kaul, Zubin Aysola, and Justin Johnson. Redcaps: Web-curated image-text data created by the people, for the people. *arXiv preprint arXiv:2111.11431*, 2021. 6

[11] Jian Ding, Nan Xue, Gui-Song Xia, and Dengxin Dai. Decoupling zero-shot semantic segmentation. In *CVPR*, 2022. 3, 7, 8

[12] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2020. 5

[13] Pinar Duygulu, Kobus Barnard, Joao FG de Freitas, and David A Forsyth. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In *ECCV*, 2002. 2

[14] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 2009. 6

[15] Golnaz Ghiasi, Xiuye Gu, Yin Cui, and Tsung-Yi Lin. Scaling open-vocabulary image segmentation with image-level labels. In *ECCV*, 2022. 2, 3, 5, 7

[16] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *NeurIPS*, 2020. 6

[17] Tanmay Gupta, Arash Vahdat, Gal Chechik, Xiaodong Yang, Jan Kautz, and Derek Hoiem. Contrastive learning for weakly supervised phrase grounding. In *ECCV*, 2020. 3

[18] Cong Han, Yujie Zhong, Dengjie Li, Kai Han, and Lin Ma. Open-vocabulary semantic segmentation with decoupled one-pass network. In *ICCV*, 2023. 2, 3, 7, 8

[19] Kunyang Han, Yong Liu, Jun Hao Liew, Henghui Ding, Jiajun Liu, Yitong Wang, Yansong Tang, Yujiu Yang, Jiashi Feng, Yao Zhao, and Yunchao Wei. Global knowledge calibration for fast open-vocabulary segmentation. In *ICCV*, 2023. 2, 3, 7, 8

[20] 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. 2021. 2

[21] Tapas Kanungo, David M Mount, Nathan S Netanyahu, Christine D Piatko, Ruth Silverman, and Angela Y Wu. An efficient k-means clustering algorithm: Analysis and implementation. *IEEE TPAMI*, 2002. 5

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

[23] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. *arXiv:2304.02643*, 2023. 3

[24] Boyi Li, Kilian Q Weinberger, Serge Belongie, Vladlen Koltun, and René Ranftl. Language-driven semantic segmentation. In *ICLR*, 2022. 2, 3, 7

[25] Feng Li, Hao Zhang, Peize Sun, Xueyan Zou, Shilong Liu, Jianwei Yang, Chunyuan Li, Lei Zhang, and Jianfeng Gao. Semantic-sam: Segment and recognize anything at any granularity. *arXiv preprint arXiv:2307.04767*, 2023. 3

[26] Yanghao Li, Haoqi Fan, Ronghang Hu, Christoph Feichtenhofer, and Kaiming He. Scaling language-image pre-training via masking. 2023. 5

[27] Feng Liang, Bichen Wu, Xiaoliang Dai, Kunpeng Li, Yinan Zhao, Hang Zhang, Peizhao Zhang, Peter Vajda, and Diana Marculescu. Open-vocabulary semantic segmentation with mask-adapted clip. In *CVPR*, 2023. 2, 3, 7

[28] 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 *ECCV*, 2014. 6

[29] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017. 5

[30] 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 *ECCV*, 2022. 7

[31] Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob

Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. In *NeurIPS*, 2020. 3

[32] Huaishao Luo, Junwei Bao, Youzheng Wu, Xiaodong He, and Tianrui Li. Segclip: Patch aggregation with learnable centers for open-vocabulary semantic segmentation. 2023. 2, 7

[33] Fangyun Wei Han Hu Xiang Bai Mengde Xu, Zheng Zhang. Side adapter network for open-vocabulary semantic segmentation. 2023. 2, 3, 7

[34] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *Proc. 3DV*, 2016. 5

[35] 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 *CVPR*, 2014. 6

[36] Norman Mu, Alexander Kirillov, David Wagner, and Saining Xie. Slip: Self-supervision meets language-image pre-training. In *ECCV*, 2022. 5

[37] 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. 2021. 1, 2, 4, 5, 6, 7

[38] Kanchana Ranasinghe, Brandon McKinzie, Sachin Ravi, Yinfei Yang, Alexander Toshev, and Jonathon Shlens. Perceptual grouping in vision-language models. In *ICCV*, 2023. 2, 4, 7

[39] Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, et al. Grounded sam: Assembling open-world models for diverse visual tasks. *arXiv preprint arXiv:2401.14159*, 2024. 3

[40] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2018. 2, 6

[41] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE TPAMI*, 2000. 5

[42] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 2017. 5, 6

[43] 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. In *CVPR*, 2024. 7

[44] Weijia Wu, Yuzhong Zhao, Mike Zheng Shou, Hong Zhou, and Chunhua Shen. Diffumask: Synthesizing images with pixel-level annotations for semantic segmentation using diffusion models. In *ICCV*, 2023. 2, 3

[45] Bin Xie, Jiale Cao, Jin Xie, Fahad Shahbaz Khan, and Yanwei Pang. Sed: A simple encoder-decoder for open-vocabulary semantic segmentation. In *CVPR*, 2024. 7

[46] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *CVPR*, 2020. 6

[47] 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, 3, 4, 5, 6, 7, 8

[48] 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 *CVPR*, 2023. 1, 2, 3, 7

[49] 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 *CVPR*, 2023. 2, 7

[50] Qihang Yu, Ju He, Xueqing Deng, Xiaohui Shen, and Liang-Chieh Chen. Convolutions die hard: Open-vocabulary segmentation with single frozen convolutional clip. *Advances in Neural Information Processing Systems*, 36, 2024. 7

[51] Haobo Yuan, Xiangtai Li, Chong Zhou, Yining Li, Kai Chen, and Chen Change Loy. Open-vocabulary sam: Segment and recognize twenty-thousand classes interactively. *arXiv preprint arXiv:2401.02955*, 2024. 3

[52] Chong Zhou, Chen Change Loy, and Bo Dai. Extract free dense labels from clip. In *ECCV*, 2022. 2, 4, 6, 7

[53] Barret Zoph, Golnaz Ghiasi, Tsung-Yi Lin, Yin Cui, Hanxiao Liu, Ekin Dogus Cubuk, and Quoc Le. Rethinking pre-training and self-training. *NeurIPS*, 2020. 6