Open-Vocabulary Image Segmentation

Golnaz Ghiasi, Xiuye Gu, Yin Cui, Tsung-Yi Lin
Google Research
{golnazg,xiuyegu,yincui,tsungyi}@google.com

Abstract

We design an open-vocabulary image segmentation model to organize an image into meaningful regions indicated by arbitrary texts. We identify that recent open-vocabulary models [22, 38] can not localize visual concepts well despite recognizing what are in an image. We argue that these models miss an important step of visual grouping, which organizes pixels into groups before learning visual-semantic alignments. We propose OpenSeg to address the above issue. First, it learns to propose segmentation masks for possible organizations. Then it learns visual-semantic alignments by aligning each word in a caption to one or a few predicted masks. We find the mask representations are the key to support learning from captions, making it possible to scale up the dataset and vocabulary sizes. Our work is the first to perform zero-shot transfer on holdout segmentation datasets. We set up two strong baselines by applying class activation maps or fine-tuning with pixel-wise labels on a pre-trained ALIGN model [22]. OpenSeg outperforms these baselines by 3.4 mIoU on PASCAL-Context (459 classes) and 2.7 mIoU on ADE-20k (847 classes).

1. Introduction

Image segmentation is an important step to organize an image into a small number of regions in order to understand “what” and “where” are in an image. Each region represents a semantically meaningful entity, which can be a thing (e.g., a chair) or stuff (e.g., floor). Language is a natural interface to describe what is in an image. However, semantic segmentation algorithms often only learn with closed-set categories, and thus are unable to recognize concepts outside labeled datasets. Figure 1 shows examples of image segmentation driven by language. The segmentation model takes text queries as inputs and produces segmented regions with arbitrary text queries.

Recently, CLIP [38] and ALIGN [22] learn with billion-scale image-text training examples to understand “what” are in an image with arbitrary text queries. These models demonstrate impressive results when directly evaluated on downstream image-text retrieval or classification tasks. However, localizing text queries to understand “where” these visual concepts in an image is still challenging.
example, Figure 2 shows the segmentation predictions of a pre-trained ALIGN [22] model using class activation maps [53].

We argue that what is missing in these state-of-the-art open-vocabulary models are mid-level representations from visual groupings [46], which organizes an image into a small set of segmentation masks. Furthermore, visual-semantic alignments should perform after grouping to align texts to segmentation regions. However, these models represent an image with a single feature vector, inevitably losing much location information.

An alternative approach towards open-vocabulary image segmentation is zero-shot image segmentation [4, 6, 19, 28, 47]. Instead of representing an image with a feature vector, these methods typically represent an image with a high resolution feature map, and learn with pixel-wise class labels. However, the issue of this approach is in the scalability of training data. It is costly to annotate pixel-wise class labels, and thus requires generalization to unseen visual concepts from limited class labels.

In this work, we represent an image with a set of segmentation masks and their features. We implement a class-agnostic segmentation module with region-to-image cross-attention [8, 10, 44] and train it with class-agnostic segmentation masks. In contrast to the works using similar architectures [10, 44], we do not predict the “no object” label ә to indicate if a predicted mask is a valid group of pixels. Considering the training data is only annotated with one possible organization of an image, we allow our model to predict other possible organizations beyond the annotations present in the training datasets.

Next, we learn visual-semantic alignments based on the predicted masks, which provide two major benefits in training. First, we perform mask-based feature pooling to aggregate pixels inside the predicted mask to generate location-aware region features. Second, the small number of predicted masks makes it easier to learn weakly-supervised alignments between regions to words in an image caption. The ability to learn from weak labels is important for scaling up training data and increasing vocabulary sizes. We call our method OpenSeg, standing for open-vocabulary image segmentation.

To evaluate our method, we measure performances on holdout image segmentation datasets. We want to promote the framework where the model can be trained with a large scale supervised/weakly-supervised data to learn generalist models transferable to other datasets. Such a scenario has been recently developed and popularized for image classification [22, 38] and object detection [17, 50]. To our knowledge, OpenSeg is the first work in image segmentation to demonstrate zero-shot transfer ability across datasets using language. This is in stark contrast to the existing evaluation protocol which measures performances of specialist models trained and tested using limited labeled data from the same dataset distribution.

In our experiments, we train the mask prediction model using class-agnostic mask annotations in the Panoptic COCO dataset [25]. We show that the model can generalize well to other datasets, reaching superior performances compared with prior works on segmentation proposals [3, 31]. Then, we report mean intersection over union (mIoU) metrics for measuring both localization quality and accuracy of open-vocabulary semantic recognition. We set up two strong baselines by applying class activation maps [53] or fine-tuning with pixel-wise labels on a pre-trained ALIGN model [22]. These baselines have already surpassed the performances in existing zero-shot image segmentation methods [4, 6, 19, 28, 47] on several datasets due to large-scale image-text pre-training. OpenSeg outperforms the baselines by 3.4 mIoU on PASCAL-Context (459 categories) and 2.7 mIoU on ADE-20k (847 categories).

2. Related Work

Grouping for visual recognition: Grouping has been a core research area in mid-level visual representations. The importance of grouping for human perception was pointed out almost a hundred years ago [46]. In machine perception, early works [11, 40] group pixels based on local affinities. Arbelaez et al. find contour detection and multiscale information helpful to generate segmentation [2] and use it to predict object candidates [3]. COB [31] improves the efficiency and performance by leveraging deep nets. These mid-level region representations are then used for semantic segmentation [31] and object detection [43]. Recently, Qi et al. [37] propose to segment all visual entities without considering semantic category labels and show generalization to unseen domains. In contrast to [37], our work aims to predict segmentation and understand the semantics
of segmented regions by open-vocabulary visual-semantic alignments.

**Fully-supervised segmentation:** To understand semantics of pixels, several datasets have been developed with an increasing number of images and categories [5, 7, 12, 33]. Models trained and evaluated on these datasets can only learn to recognize the pre-defined classes, which are at most in the order of thousands [5, 7, 33]. However, most classes only have one or few examples, thus the community often works on learning a subset of hundred classes. The classes across datasets are not transferable. MSeg [27] points out the ambiguity of class definitions, and shows that it requires manual efforts to clean up the ambiguity in order to learn a transferable model across datasets. But the model still can not transfer to new visual concepts not present in supervised datasets. Zhao et al. [52] propose to use WordNet hierarchy to aid knowledge transfer. On the other hand, our work aims at learning a model that can take arbitrary text queries for image segmentation.

**Semantic segmentation with less supervision:** Approaches using less supervision are more scalable. Weakly-supervised semantic segmentation can train with image-level labels [23, 29, 34, 45, 49], of which refining CAMs [53] is a popular techniques. Zero-shot semantic segmentation methods [4, 6, 19, 28, 47] aim to segment images with unseen visual concepts using language embeddings. These approaches learn with pixel-wise class labels which are expensive to scale up due to the long-tailed nature mentioned in the previous paragraph. Nonetheless, most previous works are evaluated on limited number of categories. On the contrary, we evaluate on datasets with large number of categories to verify the zero-shot transfer ability on large vocabulary.

**Visual grounding:** Image captioning and image-text coreference datasets [9, 26, 35] enable research on the interplay of captions and grounded visual concepts [13, 14, 18, 24, 39]. However, these methods often rely on an object detector to predict object bounding boxes for grounding and are not able to handle stuff. Our method also uses captions as semantically-rich supervision. We draw inspiration from these works and expand the model’s ability to ground visual concepts of both things and stuff to pixels with our mask representations.

### 3. Method

Figure 3 shows an overview of our approach. In contrast to approaches that represent an image with a vector $Z \in \mathbb{R}^{1 \times D}$ or a feature map $Z \in \mathbb{R}^{H \times W \times D}$, OpenSeg represents an image with $N$ proposal masks with their features $Z \in \mathbb{R}^{N \times D}$. Our mask representations support learning precise image segmentation with image captions by weakly-supervised learning. In Section 3.1, we describe the learning of predicting mask proposals from an image. In Section 3.2, we describe the feature representations of proposal and the learning of region-word alignments. In the following sections, We use a bold symbol to indicate an array of elements $\mathbf{x} = \{x_1, x_2, ..., x_n\}$, where the first dimension indicates the number of elements.

#### 3.1. Learning Segmentation Masks

We design a model architecture which consists of a feature pyramid network (FPN) [30] for multi-scale feature extraction and a cross-attention module for segmentation region proposal. We fuse FPN features into $P_2$ resolution as described in [16] to generate image features $\mathcal{F}$. We obtain $\mathcal{F}_x \in \mathbb{R}^{H \times W \times D}$ by convolution and fc layers fol-
allowed by \( \mathcal{F} \). Then we augment image features by adding learnable position embeddings \( PE: \mathcal{F}_\mathcal{P} = \mathcal{F}_s + \mathcal{F}_s + PE \). We use a cross-attention module taking inputs as \( \mathcal{F}_s \mathcal{P} \) and a randomly initialized queries \( q^0 \in \mathbb{R}^{N \times D} \) to generate mask queries \( q \in \mathbb{R}^{N \times D} \). Then, we compute the dot product of mask queries and position-augmented image features to predict masks \( s = \text{Sigmoid}(\text{dot}(q, \mathcal{F}_s \mathcal{P}))) \in \mathbb{R}^{N \times H \times W} \). This architecture is conceptually similar to Max-deeplab [44] and MaskFormer [10]. The details of the architecture are in Appendix C.

We compute Dice coefficient [32] between predicted masks \( s \) and class-agnostic labeled masks \( s' \in \mathbb{R}^{M \times H \times W} \) and maximize the Dice coefficient of the best matched mask for each labeled mask.

\[
\mathcal{L}_S = \frac{1}{M} \sum_{j=1}^{M} (1 - \max_i \text{Dice}(s_i, s'_j))
\]  

(1)

Typically, \( N > M \) for each training image. Therefore, a subset of proposal masks are optimized to best match labeled masks. The rest of proposals can still segment out unlabeled regions without being penalized. One predicted mask may match to multiple labeled masks in the early training stage when their overlaps are low. But this does not prevent learning masks that highly overlap with labeled masks in the latter training stage.

### 3.2. Visual-Semantic Alignment with Masks

We use a pair of image \( I_b \) and caption \( C_b \) to learn visual-semantic alignments. We break \( I_b \) into regions (Section 3.1) and \( C_b \) into words by extracting list of nouns from the caption. We randomly drop each word with the probability of \( 1 - kp \). We generate image features \( \mathcal{F}_z \) using the same architecture as \( \mathcal{F}_s \). For each region, we compute its feature by pooling image features with the mask \( z[n, d] = \sum_{i,j} s^{b}[n, i, j] \cdot \mathcal{F}_z[i, j, d] \). We feed each word to a pre-trained text encoder to compute the word feature \( w \).

We follow the grounding loss in prior works [18, 50] to learn region-word alignments. We first define the notation for Softmax on an array \( x \) to get the normalized score at the \( i \)-th element:

\[
\sigma(x)_i = \frac{e^{x_i/\tau}}{\sum_j e^{x_j/\tau}}
\]  

(2)

where \( \tau \) is a learnable scalar for the temperature. The similarity score of a region \( i \) and a word \( j \) is defined by its cosine similarity \( \langle z_i, w_j \rangle = \frac{z_i \cdot w_j}{\|z_i\| \cdot \|w_j\|} \). Then we define the similarity of all regions \( z \) to a word \( w \) as: \( g(z, w) = [\langle z_1, w \rangle, \ldots, \langle z_N, w \rangle] \in \mathbb{R}^{N \times 1} \). We compute the similarity of an image \( I_b \) and its caption \( C_b \) by:

\[
G(I_b, C_b) = \frac{1}{K} \sum_{j=1}^{K} \sum_{i=1}^{N} \sigma(g(z, w))_i \cdot \langle z_i, w_j \rangle
\]  

(3)

The above similarity function encourages each word to be grounded to one or a few regions. Also, it avoids penalizing regions that can not find any similar word. Next, a grounding loss is defined for a given mini-batch \( B \), where each example contains an image-caption pair. We define the similarity scores of all images in a batch \( \mathcal{I} \) to a caption \( C_b \) by \( G(I, C_b) = [G(I_1, C_b), ..., G(I_B, C_b)] \in \mathbb{R}^{|B| \times 1} \) and similarly \( G(I_b, C) = [G(I_b, C_1), ..., G(I_b, C_B)] \in \mathbb{R}^{|B| \times 1} \). The grounding loss aims at maximizing the normalized score of a labeled image-caption pair \( (I_b, C_b) \) over all images and all captions in a mini-batch.

\[
\mathcal{L}_G = -\frac{1}{|B|} \sum_{b=1}^{B} \left( \log \sigma(G(I_b, C_b))_b + \log \sigma(G(I_b, C))_b \right)
\]  

(4)

To train OpenSeg, we simply sum the two losses with a weight \( \alpha \):

\[
\mathcal{L} = \mathcal{L}_G + \alpha \mathcal{L}_S
\]  

(5)

When setting \( \alpha = 0 \), the model learns without labeled segmentation, and thus needs to induce mask predictions with the visual-semantic grounding loss. We find this setting leads to a poor performance, which suggests class-agnostic mask annotations are critical for learning mask predictions.

### 3.3. Learning from Caption Only Data

Since annotating images with segmentation is expensive, to scale up the training data we need to learn from images with only caption annotations. We follow MuST [16] and first train a teacher model on a segmentation dataset with only the segmentation loss \( \mathcal{L}_S \). Then we annotate a large image-text dataset with pseudo segmentation labels using the teacher model. Lastly, the OpenSeg model is trained with the mix of human and pseudo labels.

### 3.4. Inference

Up to this point, we learn a vision model that predicts segmentation masks \( s \in \mathbb{R}^{N \times H \times W} \) and corresponding features \( z \in \mathbb{R}^{N \times D} \). Given an evaluation segmentation dataset, we encode its categories using the text encoder. If a category is defined by more than one word, we simply include all word embeddings for that category. We obtain \( K \) word embeddings \( w \in \mathbb{R}^{K \times D} \) representing all categories. The region logits are obtained by taking the cosine similarity between words and regions \( \langle w, z \rangle \in \mathbb{R}^{K \times N} \). We multiply the region logits and segmentation masks to obtain segmentation logits at each pixel \( y = \langle w, z \rangle \cdot s \in \mathbb{R}^{K \times H \times W} \). Then the category prediction at each pixel is an argmax of segmentation logits along the word dimension:

\[
\text{pred}[i, j] = \arg\max_k y[k, i, j]
\]  

(6)
4. Experiments

4.1. Experimental Settings

Architecture. We use EfficientNet-B7 [42] as the backbone architecture and employ FPN [30] for multi-scale feature fusion. We use pyramid levels from $P_2$ to $P_5$ with feature dimension 640, upsample all feature levels to $P_2$, and then merge them by a sum operation to obtain $F$. To compute $F_z$ and $F_s$, we apply a fc layer followed by 3 layers of $3 \times 3$ convolutions with 640 channels after $F$. For text-encoder we use the frozen pre-trained BERT-Large model in ALIGN [22].

Training Parameters. All models are trained with a batch size of 1024 and image size of $640 \times 640$. We apply multi-scale jittering with a random scale between $[0.8, 1.2]$ (i.e., small scale jittering in [15]). The weight decay is set to 1e-05 and we use a learning rate 0.005 with the cosine learning rate schedule. Unless otherwise mentioned, we initialize the backbone of the model from the ALIGN checkpoint [22]. We train OpenSeg on COCO dataset for 30k steps. For training on COCO and Localized Narrative datasets, we sample examples from the datasets with equal probability and we train the model for 60k steps. We set $\alpha$ (weight of segmentation loss) to 4.0 and $kp$ (keep probability of words extracted from captions) to 0.75.

4.1.1 Training Datasets

COCO: We use the panoptic segmentation [25] and caption [9] annotations in the 2017 splits which include 118k/5k train/val images. We utilize the panoptic segmentation annotations in a class agnostic manner. When evaluating on COCO Panoptic, we treat it as a semantic segmentation dataset and our model only predicts the semantic class for each pixel.

Localized Narrative (Loc. Narr.): Localized Narrative [36] contains detailed natural language descriptions along with mouse traces for multiple datasets (COCO, Flickr, Open Images, ADE20k). We don’t train on the ADE20k portion to keep its image distribution unseen. The remaining 652k images are used for training.

4.2. Predicting Masks Across Datasets

We train the segmentation proposal model on COCO and evaluate on COCO and PC-59 with recalls at IoU 50%, 70%, and 90% as metrics. Table 1 shows performance comparisons with MCG [3] and COB [31] using their pre-computed proposals. OpenSeg shows significantly superior

Table 1. Recall of segmentation mask proposals on COCO and PASCAL-Context datasets. All methods use 128 proposals.

| Method   | COCO  | PASCAL Context-59 |
|----------|-------|-------------------|
|          | R50   | R70   | R90   | R50   | R70   | R90   |
| MCG [3]  | 41.1  | 21.4  | 4.6   | 57.8  | 31.7  | 8.7   |
| COB [31] | 46.0  | 24.8  | 4.9   | 62.9  | 37.6  | 12.1  |
| OpenSeg  | 68.9  | 48.1  | 16.9  | 84.5  | 65.1  | 29.1  |
Figure 5. (Bottom) The mIoU and Grounding mIoU results of ALIGN, ALIGN++, and OpenSeg. (Top) Segmentation predictions on an image from the ADE20k (847 categories). (First row) Predictions with all 847 classes as text queries. (Second row) Predictions with only classes in the ground-truth segmentation as text queries.

4.3. Open-vocabulary Image Segmentation

In this section, we first describe open-vocabulary baselines and our evaluation metrics. Then we discuss the experimental results with open-vocabulary baselines and state-of-the-art supervised and zero-shot methods.

ALIGN baseline: Although ALIGN [22] is trained for open-vocabulary classification, it can still roughly localizes objects and stuff with arbitrary text queries (see Figure 2). Since we initialize the backbone of OpenSeg from ALIGN’s pre-trained checkpoint, we use ALIGN as a baseline. We follow the CAM [53] method for segmentation prediction. We compute the activation map before the average pooling layer of the image encoder. Then for each spatial location we compute its cosine similarity with the text embeddings of all input categories. We assign the class with the highest similarity to each location.

ALIGN++ baseline: The above ALIGN baseline does not train on the segmentation annotations. To better adapt ALIGN to segmentation tasks, we design the ALIGN++ baseline: We add FPN and introduce high resolution map in the same approach in Section 4.1. We embed class names into text embeddings and use them as per-pixel classifiers. We fine-tune the pre-trained image encoder and FPN layers on COCO dataset using a per-pixel cross-entropy loss to align pixel embeddings with text embeddings. Figure 3(b) illustrates the model of this approach. The ALIGN and ALIGN++ baselines are methods that perform visual-semantic alignments without explicit visual grouping.

Evaluation metrics: We use two metrics, mIoU and Grounding mIoU, for evaluation. Both metrics are calculated using the standard mIoU formula [12] and only differ in the text queries for each image. The mIoU is commonly used in the literature. It measures the performance of image segmentation with fixed text queries, e.g., 847 classes when evaluated for all images in A-847. The Grounding mIoU evaluates concept grounding. An example scenario is interactive segmentation where users can specify a set of concepts in an image for the model to segment. It only uses the ground-truth classes in an image, e.g., 7 classes are used as text queries for the example in the second row of Figure 5. We show that the predictions in the mIoU and Grounding mIoU settings can look quite differently. We find that sometimes mIoU does not correctly reflect the prediction quality due to the class ambiguity, e.g., building, brick, house are all correct visual concepts to describe the object in Figure 5 but the ground-truth label is building.

Zero-shot transfer to ADE20k/PASCAL: We evaluate the performance of OpenSeg and the baselines on holdout image segmentation datasets whose train sets are not used for training. In Figure 5 (bottom), we compare ALIGN,
ALIGN++ and OpenSeg on the challenging A-847 and PC-459 datasets with large vocabularies and also on the widely used A-150 and PC-59. In the following sections we discuss our findings based on these results.

**Training on limited categories hurts generalization:** ALIGN++, which is trained with pixel-wise segmentation in COCO, outperforms ALIGN by a large margin on COCO (+35.1 mIoU) and PC-59 (+30.8 mIoU). Note COCO categories contain most of PC-59 categories. However, when we evaluate ALIGN++ on A-847 which includes a larger vocabulary, the performance of ALIGN++ drops by 0.3 mIoU and 7.7 Grounding mIoU. These results illustrate that training on the limited categories of COCO hurts the generalization of the model.

**OpenSeg improves generalization:** We compare ALIGN++ and OpenSeg in Figure 5 trained on COCO images. While OpenSeg has worse mIoU on COCO and comparable mIoU on PC-59, it generalizes better on all other benchmarks. OpenSeg outperforms ALIGN++ by +2.5 mIoU and +11.3 Grounding mIoU on A-847 and also by +1.2 mIoU and +15.0 Grounding mIoU on PC-459. The OpenSeg uses class-agnostic masks and image-level caption supervision, while ALIGN++ uses 134 per-pixel class name supervision. Although OpenSeg is trained with a weaker supervision, it has a better generalization to classes outside of COCO. These results reveal that we need open-vocabulary supervision such as captions for training a generalist model.

**Scaling training data with captions improves performance:** To scale up training data we utilize the Localized Narrative dataset, which includes detailed narratives about the objects and stuff in each image. We train a segmentation teacher model on the COCO dataset and use it to generate segmentation pseudo labels on the Loc. Narr. dataset. By scaling training data from 118k images to 652k images, the performance of OpenSeg improves on average by 2.5 mIoU and 4.8 Grounding mIoU across 4 benchmarks (see Figure 5). In Appendix F, we study the importance of using pseudo segmentation labels during training.

**Ensembling of text queries and prompt engineering:** To further improve the performance of OpenSeg we use ensembling where we include synonyms or subcategories of classes. For example, we use ‘person’, ‘child’, ‘girl’, ‘boy’, etc. for the class of ‘person’. We ensemble the multiple text queries by taking the max score as described in the Section 3.4. Also, since some of the class names of the segmentation datasets are not descriptive, we add a short context to the names. e.g. we change ‘glass’ to ‘drinking glass’. These improvements give us on average 2.6 mIoU gain across 4 datasets (see Table 2). More details are included in the Appendix H.

**Compare with existing methods:** We compare OpenSeg with state-of-the-art supervised and zero-shot segmentation methods in Table 2. Supervised methods demonstrate the upper bound performance where a specialist model is trained on the target dataset. But the trained model can not be directly evaluated on new classes or datasets. On the other hand, zero-shot methods and OpenSeg directly transfer to arbitrary classes. OpenSeg outperforms the overall mIoU of the state-of-the-art zero-shot methods on all benchmarks. However, we note for a fair comparison, models need to be evaluated with the same backbone architecture, training epochs and augmentations etc. Also, the training data and type of supervisions for OpenSeg and zero-shot work are different. So Table 2 is for reference purposes only.

Zero-shot methods aim to generalize from seen object classes with ground-truth annotations to unseen classes. Since these methods need pixel-wise supervision for seen...
Importance of backbone initialization: While the model trained from scratch is worse than initialized from a pre-trained ALIGN image encoder, it only gets slightly worse results when initialized from a NoisyStudent checkpoint. Note that all models still use the pre-trained ALIGN text encoder. The model is trained on COCO and Loc. Narr. datasets.

| A-847 | PC-459 | A-150 | PC-59 |
|-------|--------|-------|-------|
| Random init. | 4.5 | 7.6 | 18.6 | 40.6 |
| NoisyStudent init. | 6.6 | 10.7 | 24.4 | 46.9 |
| ALIGN init. | 6.8 | 11.2 | 24.8 | 45.9 |

Table 3. Backbone initialization with an ALIGN pre-trained image encoder is not critical. While the model trained from scratch is worse than initialized from a pre-trained ALIGN image encoder, it only gets slightly worse results when initialized from a NoisyStudent checkpoint. Note that all models still use the pre-trained ALIGN text encoder. The model is trained on COCO and Loc. Narr. datasets.

We are the first work which directly evaluates on holdout datasets. As a result, we outperform previous zero-shot methods.

Incorporating predicted masks at inference improves mIoU accuracy. Using the ground-truth masks can be seen as the performance upper bound when segmentation masks are perfectly predicted. The model is trained on COCO.

| A-847 | PC-459 | A-150 | PC-59 |
|-------|--------|-------|-------|
| OpenSeg - pred. masks | 6.3 | 9.0 | 21.1 | 42.1 |
| + gt. masks | (1.7) 4.6 | (3.1) 5.9 | (4.7) 16.4 | (10.0) 32.1 |

Table 4. Incorporating predicted masks at inference improves mIoU accuracy. Using the ground-truth masks can be seen as the performance upper bound when segmentation masks are perfectly predicted. The model is trained on COCO.

Using all words in training captions hurts performance. We show the mIoU performances with different text filtering to break a training caption into words. Using nouns only for training achieves the best results. The model is trained on COCO.

| A-847 | PC-459 | A-150 | PC-59 |
|-------|--------|-------|-------|
| caption filter | all words | 8.3 | 8.8 | 20.0 | 41.3 |
| noun + adj. + verb | 6.0 | 8.8 | 20.9 | 41.7 |
| noun | 6.3 | 9.0 | 21.1 | 42.1 |

Table 5. Using all words in training captions hurts performance. We show the mIoU performances with different text filtering to break a training caption into words. Using nouns only for training achieves the best results. The model is trained on COCO.

4.4. Ablation Experiments

Importance of backbone initialization: In order to save the computation, we initialize OpenSeg from the state-of-the-art ALIGN checkpoint trained on 1.8 billion examples for image-text alignments. In this section, we are curious if we can learn word-region alignments from scratch and explore the importance of initialization of the vision backbone from this checkpoint. In Table 3, we compare the performance of training OpenSeg from scratch, initializing from the NoisyStudent checkpoint [48] and initializing from the ALIGN checkpoint. For training these models, we use the same hyper-parameters, and only tune the learning rate (0.32 for scratch, 0.08 for NoisyStudent init. and 0.005 for ALIGN init.) and number of steps (180k steps for scratch and 60k for NoisyStudent and ALIGN init.).

Table 3 shows that using the NoisyStudent checkpoint to initialize the backbone achieves slightly worse results (less than 0.5 mIoU on all benchmarks) compared to using the ALIGN checkpoint. It shows initializing from the ALIGN model is not necessary for good word-region alignments. However, training from scratch is still trailing behind. We may be able to reduce the gap by increasing the batch size and training with more data. The major challenge is that OpenSeg learns from higher resolution images which make scaling data expensive.

Incorporating proposals at inference time improves accuracy: We are curious about the importance of mask proposals in OpenSeg during inference. To study this problem, we take the feature map $F_z$ in OpenSeg and perform per-pixel segmentation by taking the dot product of $F_z$ with word embeddings w. This method performs inference without mask proposals. In Table 4, we compare the performance of OpenSeg and its counterparts that do not use mask proposals (the above method) or using ground-truth as mask proposals. The performance of OpenSeg is much worse if not using proposals: mIoU on PC-59 drops from 42.1 to 32.1 and from 21.1 to 16.4 on A-150. Using ground-truth as proposals can be seen as an upper bound when we have perfect class-agnostic localization. The results show the room for improving localization. It also demonstrates even with perfect localization, the semantic alignment is still challenging.

Importance of text filtering: We train OpenSeg with image captions which may include words that do not represent any regions in an image. These noises make training more challenging. We perform a simple pre-processing on the captions and extract the list of nouns. This removes conjunctions, pronouns, adverbs, verbs, etc. which decreases the noises. In Table 5, we study the performance of OpenSeg when using different types of filtering on the captions. Keeping only nouns gets the best results. The worst results are from using all words, which show 0.2-1.1 worse mIoU. The small performance differences show OpenSeg is robust to the noise in the input words to some degree.

5. Conclusion

We propose OpenSeg, an open-vocabulary image segmentation model, to organize an image into regions with arbitrary text queries. This is in stark contrast to previous works in semantic segmentation learned to predict categories in closed vocabulary. We propose to represent an image with a set of mask regions followed by visual-semantic alignments. Such representations support weakly-supervised learning for grounding words in a caption to predicted proposals, and thus make the training data scalable. We are the first work which directly evaluates on holdout
image segmentation datasets, attaining significant performance gains against strong baselines initialized by a pre-trained ALIGN model. We hope to encourage future works to learn a generalist segmentation model that can transfer across datasets using language as the interface.

Acknowledgement

We thank Mike Mozer, Jordi Pont-Tuset, Weicheng Kuo, Quoc V. Le, Bowen Cheng for the insightful discussion.

References

[1] Sandhini Agarwal, Gretchen Krueger, Jack Clark, Alec Radford, Jong Wook Kim, and Miles Brundage. Evaluating CLIP: towards characterization of broader capabilities and downstream implications. CoRR, abs/2108.02818, 2021. 11
[2] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. PAMI, 2010. 2
[3] Pablo Arbelaez, Jordi Pont-Tuset, Jonathan T. Barron, Ferran Marques, and Jitendra Malik. Multiscale combinatorial grouping. In CVPR, 2014. 2, 5
[4] Donghyeon Baek, Youngmin Oh, and Bumsub Ham. Exploiting a joint embedding space for generalized zero-shot semantic segmentation. In ICCV, 2021. 2, 3, 7
[5] Xavier Puig Tete Xiao Sanja Fidler Adela Barriuso Bolei Zhou, Hang Zhao and Antonio Torralba. Semantic understanding of scenes through ade20k dataset. IJCV, 2018. 3
[6] Maxime Bucher, Tuan-Hung Vu, Matthieu Cord, and Patrick Pérez. Zero-shot semantic segmentation. NeurIPS, 2019. 2, 3, 7
[7] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Coco-stuff: Thing and stuff classes in context. 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 Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, ECCV, 2020. 2
[9] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO captions: Data collection and evaluation server. CoRR, abs/1504.00325, 2015. 3, 5
[10] Bowen Cheng, Alexander G. Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. arXiv, 2021. 2, 4, 5, 7, 11
[11] Dorin Comaniciu and Peter Meer. Robust analysis of feature spaces: Color image segmentation. In CVPR, 1997. 2
[12] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. IJCV, 2010. 3, 5, 6
[13] H. Fang, S. Gupta, F. Iandola, R. K. Srivastava, L. Deng, P. Dollar, J. Gao, X. He, M. Mitchell, J. C. Platt, C. Zitnick, and G. Zweig. From captions to visual concepts and back. In CVPR, 2015. 3
[14] Akira Fukui, Dong Huk Park, Daylen Yang, Anna Rohrbach, Trevor Darrell, and Marcus Rohrbach. Multimodal compact bilinear pooling for visual question answering and visual grounding. arXiv preprint arXiv:1606.01847, 2016. 3
[15] Golnaz Ghiasi, Yin Cui, Aravind Srinivas, Rui Qian, Tsung-Yi Lin, Ekin D Cubuk, Quoc V Le, and Barret Zoph. Simple copy-paste is a strong data augmentation method for instance segmentation. In CVPR, 2021. 5
[16] Golnaz Ghiasi, Barret Zoph, Ekin D Cubuk, Quoc V Le, and Tsung-Yi Lin. Multi-task self-training for learning general representations. In ICCV, 2021. 3, 4
[17] Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. Zero-shot detection via vision and language knowledge distillation. CoRR, abs/2104.13921, 2021. 2
[18] Tanmay Gupta, Arash Vahdat, Gal Chechik, Xiaodong Yang, Jan Kautz, and Derek Hoiem. Contrastive learning for weakly supervised phrase grounding. In ECCV, 2020. 3, 4
[19] Ping Hu, Stan Sclaroff, and Kate Saenko. Uncertainty-aware learning for zero-shot semantic segmentation. In NeurIPS, 2020. 2, 3
[20] Shihua Huang, Zhichao Lu, Ran Cheng, and Cheng He. FaPN: Feature-aligned pyramid network for dense image prediction. In ICCV, 2021. 7
[21] Ye Huang, Wenjing Jia, Xiangjian He, Liu Liu, Yuxin Li, and Dacheng Tao. Channelized axial attention for semantic segmentation. arXiv preprint arXiv:2101.07434, 2021. 7
[22] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V Le, Yuhuan Sun, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. ICML, 2021. 1, 2, 3, 5, 6
[23] Sanghyun Jo and In-Jae Yu. Puzzle-cam: Improved localization via matching partial and full features. arXiv preprint arXiv:2101.11253, 2021. 3
[24] Aishwarya Kamath, Mannat Singh, Yann LeCun, Ishan Misra, Gabriel Synnaeve, and Nicolas Carion. Mdeetr-modulated detection for end-to-end multi-modal understanding. arXiv preprint arXiv:2104.12763, 2021. 3
[25] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollar. Panoptic segmentation. In CVPR, 2019. 2, 5
[26] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. IJCV, 2017. 3
[27] John Lambert, Zhuang Liu, Ozan Sener, James Hays, and Vladlen Koltun. MSeg: A composite dataset for multi-domain semantic segmentation. In CVPR, 2020. 3, 6, 11
[28] Peike Li, Yunchao Wei, and Yi Yang. Consistent structural relation learning for zero-shot segmentation. NeurIPS, 2020. 2, 3, 7
[29] Yi Li, Zhanghui Kuang, Liyang Liu, Yimin Chen, and Wayne Zhang. Pseudo-mask matters in weakly-supervised semantic segmentation. In ICCV, 2021. 3
A. Potential societal impacts

OpenSeg provides a language interface for recognition and localization of visual concepts. Such interface may facilitate downstream applications such as interactive intelligent home assistants, interactive content creation, and instructing robot actions with language [41]. However, OpenSeg models are trained on large-scale datasets, which may contain bias towards certain image and text distributions. As a result, we share the similar concern as recent studies for evaluating image-text models like Clip [1]. Thus OpenSeg is not suitable for deployment in the real world without properly studying the model biases and calibrating the predictions.

B. Limitations of our approach

The limitations come with the strength of our method. The mask representations can organize an image into a small number of regions, and thus enable scalable visual-semantic alignments. However, our method will not be able to segment visual concepts that do not have associated segmentation proposals. That being said, we still believe the generalization to unseen concepts with class-agnostic mask representations is easier than pixel-wise representations. How to improve generalization or adaptation of mask predictions will be an interesting research topic.

C. Architecture of the cross-attention module

Figure 6 shows the model architecture of the cross-attention module. The mask queries interact with the position-augmented image features $F_j + PE$ to generate mask queries $q^{(t+1)}$. The first mask queries $q^{(0)}$ is randomly initialized at the beginning of training. The module is stacked three times ($T = 3$) in our experiments. We try to include self-attention of queries followed by the query-image cross-attention as in [10], but that does not improve performance.

D. Mask generalization on MSeg dataset

In this experiments, we use the setup and the curated annotations in MSeg [27]. Our goal is to verify if a model trained on a single dataset can generalize to multiple datasets. Table 6 summarizes the results of recall at an IoU of 0.5. The model achieves the best results when trained on MSeg, which aggregates the training images and annotations of all datasets, and can be seen as the performance upper bound. The results are slightly worse when trained on COCO, showing the model trained on COCO can generalize reasonably well. The model trained on Cityscapes generalizes poorly, indicating OpenSeg requires a large and diverse training dataset to provide class-agnostic segmentation proposals.

E. Visualization of full mask proposals

In Figure 4 of the main paper we present a subset of predicted segmentation masks in an unseen scene. Figure 7 shows all the 128 mask proposals on the same image.

F. Importance of segmentation loss

Since grounding loss $L_G$ back-propagates through the segmentation head, it is possible to train the complete OpenSeg model with only the grounding loss. In this section we study the importance of having segmentation loss in addition to the grounding loss in training OpenSeg. In Table 7, we compare the performance of OpenSeg when it is trained with only grounding loss (second row) vs when it is trained with both losses (first row). The former model has significantly worse performance which illustrates the importance of segmentation loss in learning the visual grouping.

When training with COCO + Loc. Narr., we annotate Loc. Narr. dataset with mask pseudo labels (Section 4.3), so that we can compute both grounding and segmentation losses on all training examples. We study the importance of the pseudo labels and the segmentation loss by setting the weight of segmentation loss to zero on examples from Loc. Narr. In Table 7, we compare the performance of these two approaches. The model trained with mask pseudo labels (third row) has worse mIoU on PC-459. However, it has better performance on A-847, PC-459 and A-150 datasets, which include categories outside the COCO dataset. These results indicate that adding mask pseudo labels and computing the segmentation loss on all of the training examples helps the generalization of OpenSeg.

G. Ablation on randomly dropping words

In our experiments, we extract the list of nouns from the captions and then we keep each word by the probability of 0.75. In Table 8 we compare the performance of OpenSeg with different keeping probabilities. We obtain the best results when the keep prob is 0.5 or 0.75, which shows that randomly dropping words within a certain probability range prevents overfitting and improves the performance.
| Train / Test | COCO | ADE20K | Mapillary | IDD | BDD | Cityscapes | SUN |
|-------------|------|--------|-----------|-----|-----|------------|-----|
| Cityscapes  | 28.8 | 25.8   | 42.5      | 50.0| 61.2| 69.3       | 22.4|
| COCO        | 82.2 | 68.0   | 48.6      | 58.1| 64.2| 58.6       | 90.8|
| MSeg        | 82.9 | 80.3   | 55.8      | 68.1| 73.4| 67.2       | 93.6|

Table 6. **Our mask prediction can generalize across datasets.** We report the recall at IoU 0.5.

| | Segmentation loss | | Segmentation loss | | | |
| | COCO | Loc. Narr. | | COCO | Loc. Narr. | | | |
| OpenSeg(COCO) | ✓ | - | 6.3 | 9.0 | 21.1 | 42.1 | |
| OpenSeg(COCO) | × | - | (-3.8) | (2.5) | (-5.7) | 3.3 | (-14.6) | 6.5 | (-26.5) | 15.6 | |
| OpenSeg(COCO + Loc. Narr.) | ✓ | ✓ | 6.8 | 11.2 | 24.8 | 45.9 | |
| OpenSeg(COCO + Loc. Narr.) | ✓ | × | (-0.5) | 6.3 | (-0.8) | 10.4 | (-2.4) | 22.4 | (+1.4) | 47.3 | |

Table 7. **Grounding loss is not sufficient for training OpenSeg.** We experiment with setting segmentation loss to zero on COCO examples when training on COCO, or on Loc. Narr. examples when training on COCO + Loc. Narr. In both cases the performance of the model drops.

| | A-847 | PC-459 | A-150 | PC-59 |
| | | | | |
| kp = 1.0 | 5.8 | 10.4 | 22.7 | 44.9 |
| kp = 0.75 | 6.8 | 11.2 | 24.8 | 45.9 |
| kp = 0.5 | 7.0 | 10.7 | 25.0 | 46.0 |
| kp = 0.25 | 6.8 | 9.8 | 22.4 | 43.5 |

Table 8. **Randomly dropping words improves performance of OpenSeg.** We extract the list of nouns from captions and keep each noun by a probability of kp. We get the best results with kp = 0.5/0.75.

### H. Ensembling and prompt engineering

In this section, we study how we can further improve the performance of OpenSeg by prompt engineering and ensembling with the class names provided by testing segmentation datasets.

**Ensembling:** An object can often be referred with more than one possible description. Some of them exist in the testing dataset, and some do not. For example, image captions usually include one of the descriptions of ‘man, woman, boy, girl, etc.’ when referring to the ‘person’ category in the testing segmentation datasets. Thus to further improve the performance, we manually assemble a list of synonyms, subcategories and plurals for some of the categories. Here are a few examples:

- person → person, child, girl, boy, woman, man, people, children, girls, boys, women, men
- dog → dog, puppy, dogs, puppies
- cat → cat, kitty, cats, kittens
- grass → grass, grasses, lawn, turf
- bottle → bottle, bottles, water bottle, water bottles

**Polysemy:** Some class names of the segmentation datasets are polysemous. As a result, a model may make predictions for different meanings of a concept, while the dataset only includes one of the meanings. For example, the class ‘fan’ in ADE20k refers to a cooling machine, but OpenSeg sometimes labels a crowd of fans (people) on a stage watching a game as ‘fan’. To resolve this issue, we add a short context to some of the labels. e.g., we change ‘fan’ to ‘ceiling fan, floor fan’.

**Categories have overlap:** Another challenge is that some of the classes of a dataset may have overlap. Although the annotators may follow some rules that prevent the overlap. An example of this can be seen in the Figure 5. A-847 has both ‘roof’ and ‘building’ categories, and both of them are correct labels for the ‘roof’ region in this figure. Another example is the ‘clothes’ and ‘person’ categories in the A-150 and PC-59 datasets, where a model is penalized for predicting the ‘clothes’ category on a person. This issue happens more frequently as the vocabulary size gets larger. We don’t have a good solution for this issue. However, in addition to the mIoU metric, we calculate the Grounding mIoU metric that has less of this issue.

Table 9 provides the gains that we attain by applying ensembling and prompt engineering. Overall, the improvement is less significant when we scale up training data from COCO to COCO+Loc. Narr. (+2.4 vs +1.8 on average across 4 benchmarks). Moreover, since there is less ambiguity in terms of class names for the Grounding mIoU, the improvement is smaller for this metric in comparison to mIoU (+1.2 vs +1.8 on average across 4 benchmarks when we train on COCO+Loc. Narr.). We will open source our modified list of class names used for this experiment.

### I. Visualization of segmentation predictions

In Figures 8-11, we present the predictions of OpenSeg on random images in the A-150 dataset, where the list of dataset categories are used as the query. For each
| Training data            | prompt eng. | mIoU    | Grounding mIoU |
|-------------------------|-------------|---------|----------------|
|                         |             | A-847   | PC-459 | A-150 | PC-59 | A-847 | PC-459 | A-150 | PC-59 |
| COCO                    | ✗           |  6.3     |  9.0   | 21.1  | 42.1  | 21.8  | 32.1   | 41.0  | 57.2  |
| COCO                    | ✓           | (+0.5) 6.8 | (+1.0) 10.0 | (+3.7) 24.8 | (+4.3) 46.4 | (+0.6) 22.4 | (+0.6) 32.7 | (+2.5) 43.5 | (+3.7) 60.9 |
| COCO+Loc. Narr.        | ✗           |  6.8     | 11.2   | 24.8  | 45.9  | 25.4  | 39.0   | 45.5  | 61.5  |
| COCO+Loc. Narr.        | ✓           | (+0.8) 7.6 | (+0.6) 11.8 | (+2.9) 27.7 | (+3.1) 49.0 | (+0.1) 25.5 | (+0.7) 39.7 | (+1.3) 46.8 | (+2.7) 64.2 |

Table 9. Ensembling and prompt engineering improves performance of OpenSeg.

Image, we visualize the output of OpenSeg. We also show the per-pixel prediction without incorporating the mask proposals (see Section 4.4) for comparison.

Figure 7. **Full set of predicted segmentation masks.** This model is trained to predict 128 segmentation masks.
Figure 8. **Predictions of OpenSeg on random examples in the A-150 dataset** (Part1). For each example, top left is the input image, top right is the ground-truth mask, and bottom right is the output of OpenSeg. Bottom left shows per-pixel prediction of OpenSeg without incorporating segmentation proposals. Note we only display one name in the legend for each category, but each category may include a list of names.
Figure 9. Predictions of OpenSeg on random examples in the A-150 dataset (Part2).
Figure 10. Predictions of OpenSeg on random examples in the A-150 dataset (Part 3).
Figure 11. Predictions of OpenSeg on random examples in the A-150 dataset (Part4).