IVRL semester project report: Language-based segmentation

Tran Minh Son Le
EPFL
son.le@epfl.ch

Abstract

This project tackles the task of language-based segmentation in a zero-shot and open-vocabulary setting by exploiting image-text alignment and image pixel affinities from pre-trained vision-language models. More specifically, we extract cross-attention maps from constrative and generative vision-language models and self-attention maps from self-supervised vision models. Then, we investigate three methods to use and combine the two attention mechanisms: using cross-attention to make per-pixel classification, refining the cross-attention map with self-attention scores via matrix multiplication, and using unsupervised semantic segmentation methods to generate pixel clusters which are to be classified using cluster-averaged per-pixel cross-attention scores. Through extensive experiments and ablations on four common segmentation benchmarks, we find that our implementations, despite some drawbacks, are competitive with previous work, and show the potential of leveraging off-the-shelf vision-language models. Our code can be found at https://github.com/sonalexle/langseg.

1. Introduction

Image segmentation is a long-standing problem in computer vision, and even until today, is not yet considered "solved". Among the recent methods that was proposed for this task, one of the most versatile and applicable is Segment Anything (SAM) [23], which shows near-universal segmentation abilities, even in the zero-shot setting on images in the wild. However, to achieve this level of performance, the method is trained with full supervision, i.e., with ground-truth segmentation masks. Preparing segmentation datasets with full ground-truth masks can be very expensive. Therefore, there are lines of research seeking to leverage only weak supervision (*e.g.*, image class labels [4]) or even without any supervision at all.

ReCo [42] is a step towards this purpose by relying on pre-trained self-supervised vision models, DINO [8], and contrastive vision-language models (VLMs), CLIP [35]. Inspired by this line of work, in this project, we explore the usage off-the-self pre-trained VLMs. We rely on the assumptions that self-supervised vision models capture strong semantic grouping [8] and that vision-language models capture image-text semantic relationships (*e.g.*, how a text prompt is related to an image, as learned by CLIP [35]). For the former assumption, we use Stable Diffusion (SD) [36], while we use both SD and CLIP [35] for the latter. Here, SD is a generative model trained to reconstruct images, and CLIP is a discriminative model trained to recognize image-text pairs. Intuitively, the former should be good at feature clustering for image synthesis, and the latter should assign similar image-text pairs with high similarity scores, that is, the model should know which image region has a similar semantic meaning to a given text prompt, if there is such a region.

More specifically, in this project, we investigate zero-shot open-vocabulary language-based semantic segmentation via the attention mechanism [48] from pre-trained models: cross-attention for cross-modal alignment (i.e., pixel-text similarities) and self-attention for self-similarity (i.e., pixel affinities). We obtain the cross-attention maps by extracting them from a modification [5] of the CLIP model [35]. We can also extract cross-attentions from Stable Diffusion [36] since it is a text-to-image diffusion model, but we experimentally found this approach to be suboptimal. Meanwhile, as shown in [46], the SD self-attention maps are suitable as pixel affinity descriptors, and we use them as the self-attention part mentioned in above. The motivation for using self and cross-attention is that they naturally align with our objectives of zero-shot language-based segmentation via cross-modal and uni-model alignment.

We explore three methods to use and combine self and cross-attentions for image segmentation. The first method, inspired by [58] is to simply perform classification per pixel by choosing the class with the highest cross-attention score as the predicted class. The second method, inspired by [21, 49], is to refine and improve the quality of the cross-attention map via belief propagation: similar pixels should have the same class. The self-attention mechanism

precisely captures this pixel similarity information. Hence, we multiply the cross-attention map with the self-attention map. The third method, inspired by [27], is to rely on unsupervised segmentation methods to generate pixel clusters, which are classified (labeled) by averaging the cross-attention scores of the pixels in the cluster and choosing the class with the maximum average score. Since all three methods use cross-attention, the only requirement is that we know the set of candidate classes to be segmented across the images. We conduct experiments and ablation studies on four standard semantic segmentation datasets: PASCAL VOC 2012 [14], PASCAL Context [31], COCO Stuff [29], and Cityscapes [11]. We find that our implementations of the three methods are competitive with previous work, and, while having some weaknesses, show promise for further developing the idea of combining self and cross-attention maps from pre-trained models for zero-shot segmentation.

2. Related work

Language-driven semantic segmentation. There has been a recent line of research of using text for dense vision tasks such as object detection [7] and semantic segmentation [26]. Regarding semantic segmentation, many recent methods demonstrate excellent results via fine-tuning on pre-trained models [55] or training from scratch [23]. However, many such methods often rely on strong supervision using ground-truth masks.

Alternatively, there is another line of research that aims to perform language-based segmentation with language-image pairing only. This is possible by leveraging features from pre-trained vision-language models such as CLIP [28] and Stable Diffusion [36] to perform self-supervised training or training-free adaptation on target datasets. Here, we briefly describe methods most relevant to this project. ReCo [42] exploit CLIP's strong retrieval abilities to construct per-class image archives, from which they compute class prototypes via co-segmentation. Similarly, OVDiff [20] use Stable Diffusion [36] to generate the archives, with which they also compute class prototypes. Alternatively, [41] replaces [42]'s co-segmentation with unsupervised saliency detectors to generate pseudo-labels. Meanwhile, [18] performs pixel-level self-supervised learning by guidance from CLIP. In this project, we explore using CLIP for the image-text alignment information.

CLIP-based zero-shot semantic segmentation. Another direction is to directly use the CLIP model for zero-shot pixel classification instead of image classification [35], but [58] find that directly using CLIP is suboptimal and they modify the last layer of its vision encoder by removing the attention mechanism. Meanwhile, [28] propose an alternative residual path [17] in the Vision Transformer [13] (ViT) vision encoder in which they perform self-attention on value features only. [5] push this idea further by generalizing the value-value attention to self-self attention (self-attention on key, value, or query features but not mixing them like the original query-key-value attention [48]). In this project, we use [5]'s method, GEM, to generate pixel-label similarity maps, which we also refer to as cross-attention maps (since pixel features attend to text features).

Segmentation with diffusion models. A recent subfield of semantic segmentation uses text-to-image diffusion models [36] to extract dense features for further fine-tuning [24, 55] or to extract attention maps for direct zero-shot usage [21, 49]. Prompt-to-prompt [19] is a seminal work which discovers cross-attention and self-attention maps of diffusion models exhibit information on object and semantic grouping. By manipulating these attention maps, image editing or controllable generation with diffusion models can be achieved [9, 15, 19, 43]. Meanwhile, [33, 52] generate synthetic datasets with mask annotations from Stable Diffusion [36]'s (SD) cross-attention layers to train segmentors. As mentioned above, OVDiff [20] synthesize archives and extract cross-attentions to compute class prototypes. On the other hand, cross-attention maps from SD can be directly used for segmentation, but refining them with SD self-attention maps can give better results [21, 49]. In this project, we also explore with extraction of cross-attention and self-attention maps from Stable Diffusion.

Unsupervised semantic segmentation and graph-based segmentation. Unsupervised semantic segmentation (USS) methods bypass the use of annotations. The main idea here is to regard segmentation as clustering, and various clustering methods are applied on deep features from self-supervised vision models such as DINO [8]. Deep Spectral Methods [30] perform spectral clustering on DINO features via normalized cuts [40]. STEGO [16] rely on feature correspondences for cluster formation. ACSeg [27] generates adaptive concepts via learnable prototypes, and the method relies on modularity [32] in graphs. More interesting for us, they propose a variant of their method that uses CLIP [35] to label the learned clusters. Finally, DiffSeg [46] find that Stable Diffusion [36] (SD) self-attention maps have good clustering properties, and they use either K-Means on these maps or an iterative merging method based on the KL distance between the self-attention maps (similar maps are merged). In this project, we adopt [46]'s method to aggregate self-attention maps across SD layers, and we use DiffSeg itself to generate clusters which are subsequently labeled by CLIP, following the idea in [27]. Meanwhile, notice that some of these methods use graph-based methods such as spectral clustering and normalized cuts [12, 40]. Similarly, random walks on graphs have also been used for deep segmentation [4].

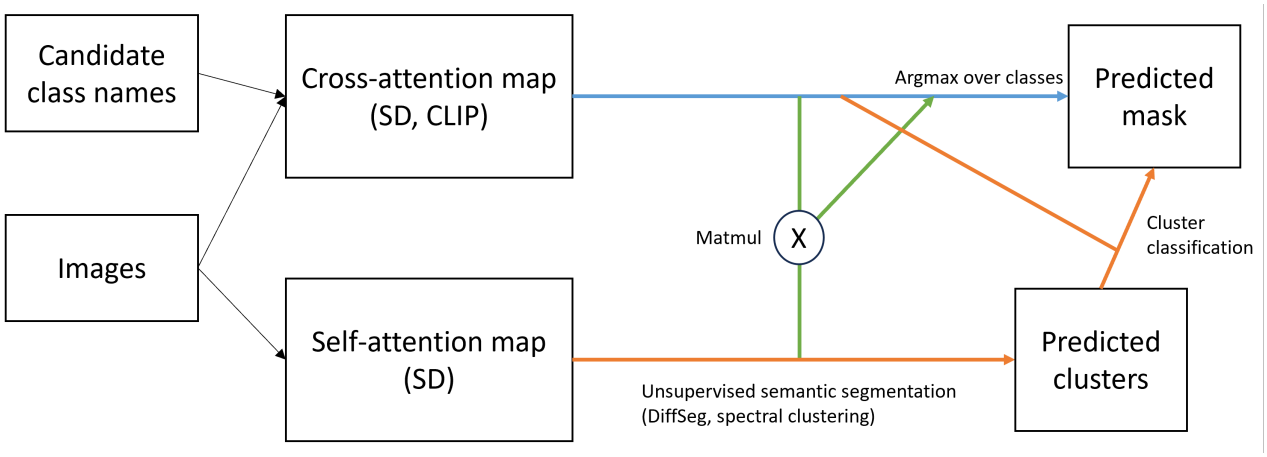


Figure 1. The schematic diagram of our implementation. Blue is method 1, green is method 2, and orange is method 3.

Another graph-based method that has been underexplored in the deep learning literature is graph cuts [6, 38]. A recent work that uses multiway cuts is [1].

3. Methodology

3.1. Preliminary: CLIP model

CLIP [35] is a discrimative vision-language model (VLM) that consists of two models, a text encoder Φ_t that encodes a text prompt P and a vision encoder Φ_v that encodes an image I. CLIP is pre-trained in a constrastive way such that the embeddings $\Phi_t(P)$ and $\Phi_v(I)$ are aligned, that is, the distance $d(\Phi_t(P), \Phi_v(I))$ is small for a matching image-text pair (I, P). Regarding the architexture, the text encoder Φ_t is a Transformer [48], and the vision encoder Φ_v can be a ResNet [17] or a Vision Transformer [13]. The ViT Φ_v consists of many transformer layers, and it transforms the image as a sequence of tokens, where each token is the embedding of an image patch. In this project, we only consider CLIP-ViT variants.

To perform image classification as well as image-text matching (during the pre-training stage), the output embedding of the special `[CLS]` image token (for a CLIP ViT image encoder) $z[0, :] \in \mathbb{R}^d$ is treated as the representation of the image, where $z \in \mathbb{R}^{(hw+1)\times d}$, hw is the number of patches, and d is the embedding dimension. Similarly, for each image class label, its representation is the output embedding of the `[EOS]` token (end of sequence) of the prompt of the label (*e.g.*, "a photo of a dog") $y \in \mathbb{R}^{c\times d}$, where c is the number of classes. Then, the class embeddings are used as a classifier by multiplying each class embedding with the image embedding to obtain their cosine similarity scores, and the argmax over the classes is the class prediction.

3.2. Preliminary: Stable Diffusion

Stable Diffusion [36] (SD) is a generative text-to-image latent diffusion model that generates images given some conditioning, which is often a text prompt. It consists of two models, a text encoder τ_θ, which is a CLIP text encoder [35], and a U-Net ϵ_θ [37]. The text encoder τ_θ encodes a prompt into token embeddings $y \in \mathbb{R}^{c\times d}$, where c is the number of text tokens and d is the hidden dimension. We explain the U-Net ϵ_θ below.

First, in latent diffusion models such as SD, the image is encoded to a latent representation $z \in \mathbb{R}^{hw\times d}$ where h, w is the spatial resolution of the latent code and d is the hidden dimension, and the diffusion processes happen in this latent space. To train a diffusion model, noise is iteratively added to z via multiple timesteps $t \in \{0 \dots T\}$ such that the resulting image after the final timestep z_T is approximately distributed as an isotropic Gaussian. This is the forward diffusion process. Then, the U-Net ϵ_θ has to predict the noise that was added to the image, given the timestep and the optional conditioning text prompt, in order to iteratively reconstruct the image from timestep T back to the original image at timestep 0. This is the reverse diffusion process. The denoised latent image $\hat{z}_0$ is finally decoded back to the pixel space $\hat{x}$.

The U-Net ϵ_θ consists of different kinds of layers, and we are interested in its attention layers. The U-Net has downsampling blocks, bottleneck blocks, and upsampling blocks. These blocks change the spatial resolution of z, and each of them contains multiple attention layers. There are two kinds of attentions in a SD U-Net, self-attention between image embeddings and cross-attention between image embeddings and text embeddings. From a self-attention layer l, we obtain the self-attention map $\mathrm{SA}^{(l)} \in \mathbb{R}^{hw\times hw}$, where h, w is the spatial resolution of the current U-Net block. Meanwhile, from a cross-attention layer l, we ob-

tain the cross-attention map $\mathrm{CA}_{\mathrm{sd}}^{(l)} \in \mathbb{R}^{hw \times c}$, where queries are computed from image features z, and keys and values are computed from text features y.

3.3. Extracting features from CLIP and SD

In this project, we extract and use dense image features from CLIP [35] (as well as CLIP text embeddings) and self and cross-attention maps from Stable Diffusion [36].

Extracting SD self and cross-attentions. As described above, each U-Net block has self and cross-attention layers. We first fix the timestep $t = 100$, which is one of the final denoising steps and here the U-Net captures fine semantic information [34]. Then, we add a corresponding amount of noise to the input image, and give as inputs to the model the text prompt, the noised input image, and the timestep t. The design of the text prompt for SD is discussed below. Next, we average the raw attention maps over the attention heads since it is found that the heads are very similar [46]. Then, we extract these maps from each attention layer l and aggregate them to generate one single cross-attention map $\mathrm{CA}_{\mathrm{sd}} \in \mathbb{R}^{hw \times c}$ and self-attention map $\mathrm{SA} \in \mathbb{R}^{hw \times hw}$, where h, w is the spatial resolution of the latent image representation z.

Extracting SD self-attentions. To obtain the aggregated SA, we use the method from [46] to aggregate across layers. In brief, for each layer l, they upsample the attention dimensions (columns of $\mathrm{SA}^{(l)}$) to the maximum resolution hw, and they compute a weighted sum of the $\mathrm{SA}^{(l)}$ maps for each spatial location (row) of the final SA. The weights of the sum are proportional to the spatial resolution of the corresponding $\mathrm{SA}^{(l)}$, and the weights sum to one. Advanced indexing is used to determine which spatial locations of the lower resolution maps correspond to those in the higher resolution maps. After the upsampling and before the sum, the rows of each $\mathrm{SA}^{(l)}$ are normalized so that they each sum to one. See [46] for a detailed description of the method.

Extracting SD cross-attentions. To aggregate the $\mathrm{CA}_{\mathrm{sd}}^{(l)}$ maps across layers l with different resolutions into a single map $\mathrm{CA}_{\mathrm{sd}}$, we simply upsample the spatial dimension of each map to the the maximum resolution hw and computing their weighted sum. Unlike the weights used to compute SA, we choose the weights here to be inversely proportional to the spatial resolution of the corresponding $\mathrm{CA}_{\mathrm{sd}}^{(l)}$. This is because the low-resolution cross-attention maps are found to capture core semantic information, while the high-resolution ones capture fine-grained details [9, 10, 19].

Extracting dense image features from CLIP. The original CLIP model [35] is trained with a global objective: image-text matching. As described above, only the `[CLS]` patch token is used for this purpose. Thus, the remaining patch tokens are not explicitly used during training [58]. One idea is to use patch embeddings as dense representations, but as investigated by [5, 28, 58], this approach is suboptimal. Instead, [58] propose to modify the CLIP ViT by removing its final self-attention layer and discarding the `[CLS]` embedding. On the other hand, [28] creates a second residual path with value-value attention layers (self-attention on value features only). Moreover, [5] improves upon this approach by generalizing value-value to key-key and query-query "self-self" attention (please see [5] for the details). They show that the self-self attention is similar to clustering, leading to better clusters and dense features. In this project, we use [5]'s method, GEM, to extract dense image features $z \in \mathbb{R}^{hw \times d}$ (excluding the `[CLS]` patch). Furthermore, given CLIP text embeddings of all class names $y \in \mathbb{R}^{c \times d}$, we can obtain a patch-text cosine similarity map by L2-normalizing the rows of z and y and computing their dot products $zy^T \in \mathbb{R}^{hw \times c}$. We can regard this map as an image-text cross-attention map $\mathrm{CA}_{\mathrm{CLIP}} = \mathrm{softmax}(zy^T/s)$ where s is a temperature ($s = 0.01$ in the original CLIP [35]).

3.4. Methods

In this section, we implement three methods. An overview is in Fig. 1.

Prompt engineering for CLIP GEM. In the setting where we have the set of all class names but not image-level captions, we need to design a prompt for the models as they observe text sequences during training instead of just a single word. For GEM [5], we follow their setup and use the prompt template `a photo of a <classname_i>` for each class $i \in \{1 \dots c\}$. Following the original CLIP [35], we take the embedding of the `[EOS]` token of the prompt as the class embedding $y[i, :] \in \mathbb{R}^d$, and concatenate across the classes to obtain the class embedding matrix $y \in \mathbb{R}^{c \times d}$.

Prompt engineering for Stable Diffusion. For SD, which uses a CLIP text encoder, [24] find that the `[EOS]` embedding method does not work well and recommend using the embedding of the `<classname_i>` tokens among all the tokens in the prompt. Hence, the SD prompt template is `a photo of <classname_1>, <classname_2>, ..., and <classname_c>`. Here, we use commas to separate between the classes as [45] find commas to disentangle the CA maps better than spaces. From the resulting raw cross-attention map, we extract only the maps of the class name tokens (instead of other tokens, *e.g.*, the comma token). Here, note that the class name string can be broken into multiple tokens. In

this case, to obtain the CA map for the class, we average over the CA maps of its tokens.

Method 1: using only cross-attention. Given a cross-attention map $\mathrm{CA}_{\mathrm{CLIP}} \in \mathbb{R}^{hw \times c}$ or $\mathrm{CA}_{\mathrm{SD}} \in \mathbb{R}^{hw \times c}$, we can directly make pixel-wise class predictions by doing an argmax over the classes c for each pixel since each $\mathrm{CA}[:, i]$ is the heatmap of the ith class. To be more specific, the predicted mask $\hat{M} \in \{0, 1\}^{hw}$ is:

$$\hat{M}[j] = \underset{c}{\mathrm{argmax}}\, \mathrm{CA}[j, :],\ j = 1 \dots hw. \tag{1}$$

Method 2: refining the cross-attention map with self-attention scores. Given a cross-attention map $\mathrm{CA} \in \mathbb{R}^{hw \times c}$ (from CLIP GEM or Stable Diffusion), we can improve its quality in terms of visual consistency and grouping by taking its matrix product with the transpose of the self-attention map $\mathrm{SA} \in \mathbb{R}^{hw \times hw}$:

$$\mathrm{AGG} = \mathrm{SA}^T \times \mathrm{CA}. \tag{2}$$

The intuition here is as follows. Consider the CA map of a class, $\mathrm{CA}[:, i] \in \mathbb{R}^{hw}$. We want to compute the weighted average of the self-attention maps, one map for each spatial location, or pixel, $\mathrm{SA}[j, :] \in \mathbb{R}^{hw}$, with its weights being the cross-attention score of that pixel $\mathrm{CA}[j, i] \in \mathbb{R}$. That is, $\mathrm{AGG}[i] = \sum_{j=1 \dots hw} \mathrm{CA}[j, i] \cdot \mathrm{SA}[j, :]$. The idea is that for a pixel j, its self-attention map $\mathrm{SA}[j, :]$ highlights other pixels similar to itself, and for a pair of pixels m, n, $\mathrm{SA}[m, :]$ and $\mathrm{SA}[n, :]$ highlight roughly the same regions if the pixels m, n have similar semantic meaning [19, 46]. Thus, the weighted sum reinforces regions and reduces the visual inconsistency of the CA activations. This aggregation method can be thought of as a random walk belief propagation, and it has been widely used, *e.g.*, to enhance Class Activation Maps (CAMs) [4] and diffusion model-based segmentation [21, 33, 49, 53]. Moreover, [58] also use this method and they call it "key denoising" as they use CLIP key features to compute the self-attention.

On the cross-attention of Stable Diffusion. We found experimentally that, when $i = 1 \dots c$ is the set of all classes, the argmax method works well for $\mathrm{CA}_{\mathrm{CLIP}}$. For $\mathrm{CA}_{\mathrm{SD}}$, we were unable to make it work, unless for each image we restrict the set of classes to the ground-truth class names. That is, SD knows exactly which classes exist in the image. We hypothesize several reasons for this failure.

Firstly, $\mathrm{CA}_{\mathrm{SD}}$, unlike $\mathrm{CA}_{\mathrm{CLIP}}$, does not natively have good calibration of the class likelihoods. More specifically, for a class token, its CA map displays the association strength of each pixel with the token. However, the association might be stronger for incorrect classes that are semantically similar to the correct class (*e.g.*, cat vs. dog). Secondly, the text prompt itself is involved in the reconstruction of the noised image in the U-Net via the cross-attention mechanism. Thus, if the text prompt contains irrelevant tokens, the reconstruction might be inaccurate, damaging the quality of the CA maps. Furthermore, since the context size of the SD text encoder is only 77 tokens, we cannot use only one prompt if there are too many classes. When the classes are split across prompts, the ordering of the class likelihoods is not immediately comparable since we found the ordering to be local for a prompt.

Therefore, in this project, when evaluating Stable Diffusion cross-attention, we only consider the weakly-supervised semantic segmentation (WSSS) setting, that is, when we know the ground-truth image-level class names. Note that this is our experimental failure, and previous works were successful with only the set of all class names [24, 49]. Note that a way to avoid using ground-truth class names is to predict the classes with a multilabel classifier. We experimented with this approach by using the CLIP-based classifier from [2] but did not obtain satisfactory results; thus, we exclude this setting from this report. To reiterate, we evaluate $\mathrm{CA}_{\mathrm{SD}}$ under WSSS, and we only use Method 2 for this. On the hand, $\mathrm{CA}_{\mathrm{CLIP}}$ is used in all methods.

Method 3: labeling clusters from self-attention with cross-attention scores. Another way to do text-driven semantic segmentation is to rely on unsupervised semantic segmentation (USS) methods. More specifically, USS methods return mask predictions which represent predicted segments, or clusters, without assigned semantic meanings (i.e., they do not predict classes). During evaluation, the Hungarian algorithm [25] is used to match predicted clusters with ground-truth clusters. This procedure could be seen as labeling predicted clusters with the class of the best-matching ground-truth cluster. Hence, we replace the Hungarian algorithm labeling with cluster-averaged predicted class scores from a cross-attention map. More specifically, given a pixel cluster K (represented as a set of pixel indices), we compute a score for each class, and we label the cluster with the class having the maximum score (argmax operation). For cluster K, the score of class i is computed by averaging the cross-attention values of the pixels in the cluster:

$$p^{(K)}[i] = \frac{1}{|K|} \sum_{j \in K} \mathrm{CA}[j, i]. \tag{3}$$

This method of classifying clusters has the advantage that if the clusters are accurate enough, the CA scores need not be pixel-perfect unlike the aggregation method above, since only the region-level average scores need to be accurate instead of pixel-level scores. On the other hand, the drawback is that if the clusters are not accurate, the errors

from the CA scores are amplified. This method has been used for diffusion-based image editing [15, 34] and USS with text [27, 46].

In this project, we explore two USS methods to generate clusters, both of which use the SD self-attention maps: spectral clustering [30] and DiffSeg [46]. For spectral clustering, the idea is to perform normalized cuts [40] on $\mathrm{SA} \in \mathbb{R}^{hw \times hw}$. Since the SA map is asymmetric (because of the softmax and the query, key, and value weights), we use the singular value decomposition [12] instead of eigenvalue decomposition on SA. The resulting left singular vectors (we use the first few vectors) are used for K-Means clustering. Here, we can regard SA as the matrix of weights of a bipartite graph [12]. Note that eigenvalue decomposition is still possible after making the SA map symmetric by averaging itself with its transpose $(\mathrm{SA} + \mathrm{SA}^T)/2$. Spectral clustering and normalized cuts have been used for deep segmentation in previous work [3, 15, 30, 44, 50].

Failed method: combining self and cross-attentions with GrabCut. GrabCut [38] is an interactive segmentation method relying on graph cuts [6]. The method works by constructing a graph for the image, where each pixel is a node, and two terminal nodes representing the background and the object. Each pixel node is connected to other pixels in its neighborhood and the two terminals. Then, the method performs a graph cut (using the max-flow min-cut algorithm) such that each pixel node is connected to only one terminal. The cut optimizes an energy objective that depends on an unary term (background/object scores) and a pairwise term (pixel affinities). Here, we notice that the unary term corresponds to a cross-attention map, and the pairwise term corresponds to a self-attention map. Hence, we replace the original terms with our CA and SA maps and perform graph cut on them. However, since the original algorithm only has two classes (foreground/background), we modify it so that the algorithm performs a one-versus-rest classification: the foreground map is the CA map of current class, and the background is the mean of the CA maps of the remaining classes. To resolve mask predictions with overlapping pixels, for these pixels we take the argmax of the CA scores over the classes. Upon intial experimentation, we found that this implementation does not work, and thus we do not develop this idea further.

	Requirements			
Model	LD	AX	UA	WS
DiffSeg [46]	✢	✗	✗	✢
ACSeg [27]	✢	✗	✓	✢
OVDiff [20]	✓	✓	✗	✗
ReCo [42]	✓	✓	✢	✗
NamedMask [41]	✓	✓	✓	✗
MaskCLIP [58]	✓	✗	✢	✢
CLIPSurgery [28]	✓	✗	✗	✢
GEM [5]	✓	✗	✗	✢
DiffSegmenter [49]	✓	✗	✗	✢
AggSD	✗	✗	✗	✓
GEM-based CA	✗	✗	✗	✢

Table 1. Techincal requirements of select methods. Language Dependency (LD), Auxiliary Images (AX), Unsupervised Adaptation (UA), Weak Supervision (WS). ✗: not required, ✢: optional, ✓: required. Table inspired by [46]. All methods in this table are open-vocabulary.

	Extra knowledge		mIoU	
Method	GT class	GT cluster	VOC	Context
GEM	✗	✗	55.8	36.5
GEMW	✓	✗	61.1	50.5
AggGEM	✗	✗	56.6	37.8
AggSD	✓	✗	57.5	43.8
AggGEMW	✓	✗	60.7	**51.2**
DiffSegClust	✗	✗	56.7	36.7
SpecClust	✗	✗	57.2	37.3
GTClust	✗	✓	**69.9**	44.9

Table 2. Ablation study on PASCAL VOC 2012 and Pascal Context (train split of both). ✗: no, ✓: yes.

	VOC		Context	
Method	AP	AR	AP	AR
GEM	78.3	90.6	50.4	67.4
AggGEM	85.9	82.2	63.1	54.0

Table 3. Mutilabel classification on PASCAL VOC 2012 and Pascal Context (train split of both). AP: average precision (averaging over class-wise precisions), AR: average recall (averaging over class-wise recalls).

4. Experiments

4.1. Implementation details

Datasets. In this project, we use four common semantic segmentation datasets, PASCAL VOC 2012 [14] (20 classes and 1 background class) (referred to as VOC), PASCAL Context [31] (59 classes, ignoring the background class) (referred to as Context), COCO-Stuff [29] (171 low-level classes and 27 mid-level classes) (referred to as COCO), and Cityscapes [11] (27 classes). We use VOC and Context (train split) for our ablation study, and all datasets (validation split) to compare with previous methods. For COCO-Stuff, we evaluate the predictions on the 27 classes, follow-

Method	Backbone	LD	ZS	VOC	Context	COCO	Cityscapes
ACSeg [27]	DINO [8] + CLIP-ViT-B/16	✓	✗	53.9	-	28.1	-
DiffSeg [46]	SD1.4	✗	✓	-	-	**43.6**	21.2
ReCo [42]	DeiT-S/16 [47] + CLIP ViT-L/14	✓	✗	34.2*	27.2	26.3	19.3
OVDiff [20]	SD1.5 + DINO + CLIP ViT- B/16	✓	✗	**69.0**	31.4	-	-
NamedMask [41]	ResNet50 + CLIP ViT-L/14	✓	✗	60.7	-	-	-
MaskCLIP [58]	CLIP ViT-B/16	✓	✓	29.1*	25.5	-	22.7@
CLIPSurgery [28]	CLIP ViT-B/16	✓	✓	41.2+	29.3	-	**31.4**
GEM [5]	MetaCLIP ViT-B/16	✓	✓	46.8	34.5	-	-
DiffSegmenter [49]	SD1.5	✓	✓	60.1	27.5	-	-
GEM (our impl.)	MetaCLIP ViT-B/16	✓	✓	55.7	37.3	38.2	10.4
AggGEM	MetaCLIP ViT-B/16 + SD1.5	✓	✓	56.7	**38.6**	39.6	7.5
DiffSegClust	MetaCLIP ViT-B/16 + SD1.5	✓	✓	56.0	37.4	38.7	7.6
SpecClust	MetaCLIP ViT-B/16 + SD1.5	✓	✓	57.1	38.3	39.3	8.3
AggSD	SD1.5	✓	✓	58.3	45.1	36.4	11.2

Table 4. Benchmarking. *Results from [41]. +Results from [5]. @Results from [28]. LD: language dependency, ZS: zero-shot. ✗: no, ✓: yes. Results for AggSD are included for reference, and notice that for COCO Stuff, AggGEM is better than AggSD despite the latter having access to GT classes.

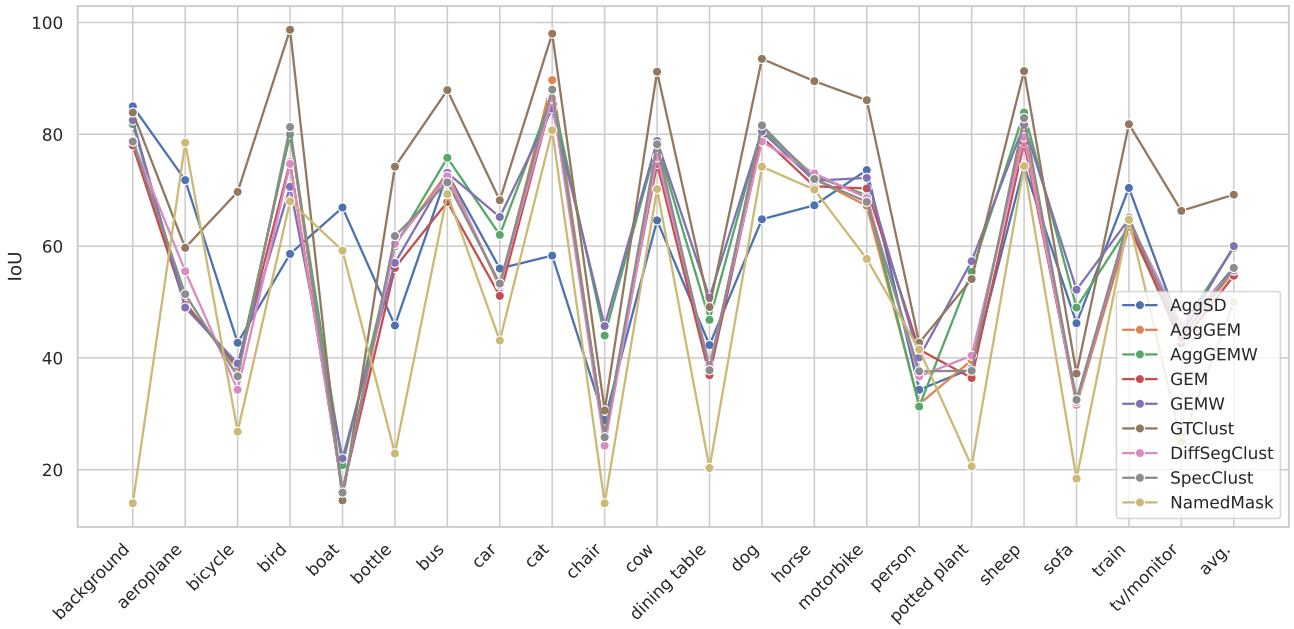


Figure 2. Class IoUs on VOC 2012 train split. Note that NameMask [41] did not report the background IoU, so we just use a random value there. Here, the avg. is computed over the class IoUs excluding the background class. Figure inspired by [22, 41].

ing previous work [27, 42, 46], but we make our models predict among the 171 classes. To be more specific, the models receive as input the 171 low-level class names and predicts masks with these class names. Then, the predicted masks (as well as the ground-truth masks) are mapped to the corresponding 27 mid-level classes. This is similar to [42]'s implementation, where they also compute prototypes for the 171 classes. [27] also follows this implementation. Regarding image preprocessing, we adopt [42]'s codebase but we do not crop; instead, we just resize the original images.

Models. In this project, we use Stable Diffusion version 1.5 [36], whose text encoder is from the CLIP ViT-L/14. For CLIP and the GEM modification [5], we use the MetaCLIP-400m ViT-B/16 model [54] since [5] find this variant to have the best performance. Regarding input image sizes, for SD, we resize input images to 512×512 pixels as this is SD 1.5's native resolution, and for GEM, we resize to 448×448 pixels following [5]. Regarding mask predictions, we evaluate on masks with resolution 320×320 following [16, 42]. Note that this means unlike previous works, our images and masks have different resolutions. Also note that SD 1.5 returns attentions maps with resolution 64×64, and we upscale them to 320×320 before making predictions via taking the argmax or labeling the clusters. Similarly, GEM returns cross-attention maps of size 28×28, and we upscale them to 64×64 to merge with the SD self-attention (Method 2), and to 320×320 to take the argmax or for cluster labeling.

Model naming. Regarding the three methods, for convenience we refer to them as follows. First, as mentioned in Sec. 3.3, for SD we use its cross-attention using Method 2 only (with ground-truth image-level class names) as we find it to work the best. This approach is referred to as AggSD. On the other hand, for GEM, we use the SD self-attention to combine with the cross-attention from GEM, and we evaluate all three methods as explained in Sec. 3. GEM with Method 1 is called GEM; GEM with Method 2 is called AggGEM; GEM with Method 3 and DiffSeg clustering [46] is called DiffSegClust; GEM with Method 3 and spectral clustering is called SpecClust.

Model details. For Method 3, whenever DiffSeg [46] is used for clustering, we use the default hyperparameters except for the timestep and the merging threshold where we set $t = 100$ and threshold of 0.8. For spectral clustering, we use 20 left singular vectors and 20 clusters for K-Means. For Method 2, the softmax temperature s used to compute $\mathrm{CA}_{\mathrm{CLIP}}$ is set to $s = 0.1$ for the VOC 2012 dataset and $s = 0.01$ for the other datasets.

Background class. For VOC 2012 dataset, we compute the background score for each pixel j as $\mathrm{BG}[j] = 1 - \max_c \mathrm{CA}_{\mathrm{CLIP}}[j] - 0.8$ for GEM (idea adopted from [57]). The resulting background map is concatenated to $\mathrm{CA}_{\mathrm{CLIP}}$ as the CA map of the background class. For AggSD, [10] find that the cross-attention map of the `[SOS]` (start of sequence) token highlights background regions or regions not referred to in the prompt. Hence, we use this map as the CA map of the background class.

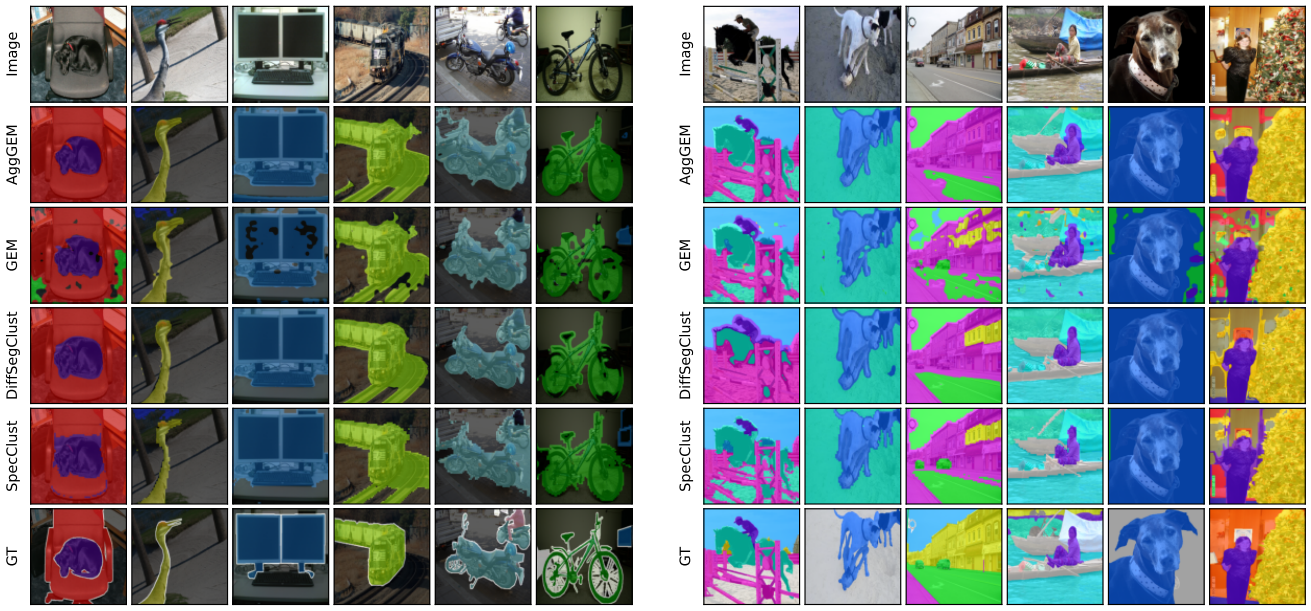
Metrics and model requirements. We use the mean intersection over union (mIoU) as our evaluation metric. In addition, following previous work [5, 46], we list the technical requirements of the methods we compare against in Tab. 1. Following [46], we consider language dependency (LD, the method requires language input as text prompts), auxiliary images (AX, the method requires an archive of reference images, real or synthetic), unsupervised adaptation (UA), as well as weak supervision (WS, image-level ground-truth class names are known). Notice that all methods listed in Tab. 1 are open-vocabulary. Please note that we only compare against methods most relevant to this project (and are also state-of-the-art), for comparison results on more related methods, please see the original papers [5, 20, 27, 28, 41, 42, 46, 49, 58].

4.2. Ablation study

In this section, we study the effect of different components: CA maps, aggregation with SA maps via multiplication, labeling clusters with CA scores. Here, there are two sources of errors: the set of class names with which to argmax over and the clustering of the pixels as clear segments. In particular, we examine the case when we have access to ground-truth image-level class names (this is always the case for AggSD) and the case when we label perfect clusters, i.e., the ground-truth masks (but no access to ground-truth class names). For the former case, we use GEM and AggGEM and call them GEMW and AggGEMW (W stands for weak supervision). The latter case is referred to as GTClust. See Tab. 2 for more details.

Language-based segmentation with stronger supervision. First, we consider the scenario where we know image-level ground-truth class names (GT classes), i.e., we have access to a perfect classifier and we only need to segment these classes in the image. This means the argmax operation is performed over fewer classes, and false positives are impossible (especially for classes with similar semantic meanings, *e.g.*, cat vs. dog). For AggSD specifically, this means we know the near-perfect caption for the image (barring descriptive adjectives and verbs), and intuitively this makes it easier for the U-Net to reconstruct the noised image. On the other hand, when GT clusters (segments) are known (i.e., we have a perfect segmentor), the model only has to classify them, which is intuitively an easier task than pixel-wise argmax.

Fig. 2 (inspired by [22, 41]) shows class-wise IoUs on the VOC 2012 train split, where we include results from [41] for reference. Notice that all the IoUs are very similar across methods, and giving the model extra knowledge boosts its performance as expected, especially for challenging classes such as potted plant and chair. Meanwhile, Tab. 2 shows mIoUs on train splits of VOC 2012 and PASCAL Context. Interestingly, AggGEMW is the best for Context instead of GTClust. This might be be-



(a) PASCAL VOC 2012. Note the dark shade is the background class.

(b) PASCAL Context.

Figure 3. Qualitative results on VOC and Context train splits (512×512 resolution). Notice the smoothing effect of AggGEM vs. GEM. The models struggle to predict the background class. Image regions having similar semantic meaning to a false-positive class may be incorrectly segmented (fourth column in Fig. 3a and final column in Fig. 3b). Even when SpecClust and DiffSegClust predict reasonable segments, the cross-attention classifier might be incorrect.

cause the Context dataset has many classes, and clusters are larger than pixels, implying that misclassification for a cluster incurs a higher cost than for a pixel. On the other hand, AggSD with GT classes is worse than GEMW and AggGEMW, implying the superior performance of CLIP-based cross-attentions to those from Stable Diffusion.

Overall, the problematic component of the segmentation pipeline in this project is the classification part (i.e., the language component), as even with a perfect segmentor, the model still struggles to correctly classify the segments. We hypothesize this is because of the lack of calibration of the cross-attention probabilities, and that the CLIP backbone is pre-trained on a global level (image-text pairs) instead of region level. One potential solution here is to train the model with dense self-supervision (see *e.g.*, [51]).

Justification of Method 2. Here, we show the benefits of refining cross-attention maps with self-attentions via matrix multiplication. Quantitatively, we consider the multilabel image classification performance of GEM and AggGEM. In this setting, we only consider which classes are present in the mask predictions. Tab. 3 shows the multilabel classification results on train splits of VOC 2012 and PASCAL Context. We observe that the refinement increases the precision and decreases the recall, meaning that the model makes fewer false positive errors. Qualitatively, because of the pixel-wise argmax in GEM and AggGEM, the predicted masks are noisy as the model has low confidence in standalone pixels or small regions. Method 2 alleviates this problem by making sure that similar pixels should have similar classes. See Fig. 3 for an illustration. [58] also make a similar observation and their key denoising strategy is similar to Method 2.

4.3. Quantitative results

In this section, we discuss results on the validation splits of all datasets, as shown in Tab. 4. ZS means zero-shot, and by zero-shot we mean the capability of a model to segment an image in the wild without any training or generating auxiliary information. This is why we do not consider ReCo [42] and OVDiff [20] to be truly zero-shot as these methods need to compute class prototypes for unseen classes, whereas MaskCLIP [58] and similar methods can simply segment images given a set of candidate classes.

Regarding Cityscapes results as shown in Tab. 4, despite the high similarity of our implementation (based on GEM [5]) with CLIPSurgery [28], the results are not satisfactory. This might be because of our image preprocessing pipeline, which is based on [42]'s implementation, and [28] use a different pipeline. Furthermore, another reason (as discussed in [46]) is that the distribution of the Cityscapes dataset (cityscapes) might not be in the training dataset of

the CLIP and diffusion backbones. Regarding VOC 2012, we observe that correctly predicting the background is crucial for the performance. Among the methods compared, only [20] has a dedicated strategy to handle the background, while other methods, including our implementations, use post-hoc approaches.

Regarding COCO Stuff results as shown in Tab. 4, although DiffSegClust is based on DiffSeg [46], the original method still performs better. This is because their method has been properly tuned to generate segments that best match ground-truth segments. In our implementation, we use a lower KL threshold and later timestamp than theirs because their hyperparameters generate larger clusters whose misclassification is more costly. Instead, we choose hyperparemters such that predicted clusters are smaller; however, by doing this, there is less change of correctly matching the ground-truth shapes that [46] can do. Moreover, the original DiffSeg (and other unsupervised methods) classify the clusters via the Hungarian algorithm [25], which, unlike Method 3, does not care about the number of classes (but cares about the number of clusters).

We include results of AggSD in Tab. 4 even though it has access to GT class names. This is to compare against diffusion-based methods [20, 49]. As mentioned in Sec. 3, we were unable to successfully implement AggSD without relying on GT classes, and we leave this to future work. Interestingly, note that for COCO Stuff, AggSD is subpar compared to our other implementations which use GEM [5] as the cross-attention (CA) *without GT class names*, demonstrating once again the effectiveness of cross-attentions in CLIP compared to those from Stable Diffusion.

In summary, the results show the potential of the idea of using pre-trained vision(-language) models for zero-shot, training-free dense recognition tasks.

4.4. Qualitative results.

Fig. 3 visualizes predictions on some images from PASCAL VOC 2012 and PASCAL Context (both using the train split). Overall, the methods make similar predictions, the difference lies in how the classification is performed. First, notice the smoothing and refinement effects of Method 2 between GEM and AggGEM, making the predicted segments more visually consistent, sometimes to the point of spreading out more than necessary, see Fig. 3a first column. Next, we observe that image regions (*e.g.*, railway) that have similar semantic meaning with some classes rain might be incorrectly segmented, see the fourth column in Fig. 3a (railway patch vs train class) and the final column in Fig. 3b (wall region vs cabinet class – the correct class there should be wall).

In addition, Fig. 3 shows that the more complex the scene with small segments, the harder it is for the models to produce correct segments. One reason is that the raw cross-attention maps are spatially very small (64×64 for SD when input size is 512×512 and 28×28 for CLIP ViT-B/16 when input size is 448×448. More importantly, the main reason which was briefly mentioned above is that the backbones (SD, CLIP) that we use are not trained with localization objectives. CLIP is only trained to connect images with text, whereas SD is trained to reconstruct images. Nevertheless, these models seem to possess latent segmentation abilities, especially the self-attention mechanism in SD and the image-text association in CLIP (after modifications [5, 28, 58]).

Finally, Fig. 3b column 4 shows that when the image is not similar to the pre-training data of CLIP, the method struggles to classify the concepts, even though segments are correctly identified. We note that this image is similar to the ones in the Cityscapes dataset, which we failed to make a working implementation.

5. Conclusion

In this project, we investigated the idea of leveraging large pre-trained vision-language models for zero-shot open-vocabulary language-based semantic segmentation. We explored the idea of combining cross-attentions (class probabilities) with self-attentions (pixel affinities). Although this idea is not new (*e.g.*, [4, 6, 34, 38, 49, 58]), it demonstrates the strengths of deep features whenever they can be combined.

We learned that CLIP [35, 54] and its modifications [5, 28] has good dense classification abilities via its image-text similarity map (the cross-attention), and that Stable Diffusion [36] has good self-clustering abilities via its self-attention mechanism. We experimented with three methods: argmax of the cross-attention only, argmax of the multiplication of cross-attention with self-attention, and classification of clusters generated by two unsupervised semantic segmentation methods, DiffSeg [46] and spectral clustering on SD self-attention [30, 40]. We unsuccessfully tried graph cuts [6, 38] to combine self and cross attentions. We tried using the cross-attention mechanism of SD for segmentation, but we were unsuccessful and found that the model needs to know ground-truth image-level class labels to work.

From the experiments, we observed that the methods we investigated were competitive with previous work (barring the failure on the Cityscapes dataset [11]). However, the shortcomings of these methods are that the classification ability (i.e., the language component) is not that great, and they may confuse between classes with similar semantic meanings. Moreover, they do not yet have the ability to segment small objects and more complex scenes, and this is due to the pre-training objectives of the backbones not being the most suitable for the downstream task of dense recognition. Nevertheless, vision-language foundation models show po-

tential for zero-shot training-free semantic segmentation if the shortcomings above can be overcome.

As future work for zero-shot language-based semantic segmentation, one direction could be to introduce better pre-training or fine-tuning objectives, such as [51]. Alternatively, as [23] shows, one can just use strong supervision to train large foundation models; however, the glaring drawback is the expensive cost of annotation. Thus, we need cheap but good pre-training tasks and datasets. On the other hand, as shown in this report, the language component as well as the vision-language correspondence needs to be improved. One direction to resolve this issue is to explore stronger language models in the vision-language framework (see, *e.g.*, [39, 56]).

References

[1] Ahmed Abbas and Paul Swoboda. Combinatorial optimization for panoptic segmentation: A fully differentiable approach. *Advances in Neural Information Processing Systems*, 34:15635–15649, 2021. 3

[2] Rabab Abdelfattah, Qing Guo, Xiaoguang Li, Xiaofeng Wang, and Song Wang. Cdul: Clip-driven unsupervised learning for multi-label image classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1348–1357, 2023. 5

[3] Amit Aflalo, Shai Bagon, Tamar Kashti, and Yonina Eldar. Deepcut: Unsupervised segmentation using graph neural networks clustering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 32–41, 2023. 6

[4] Jiwoon Ahn and Suha Kwak. Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4981–4990, 2018. 1, 2, 5, 10

[5] Walid Bousselham, Felix Petersen, Vittorio Ferrari, and Hilde Kuehne. Grounding Everything: Emerging Localization Properties in Vision-Language Transformers. *arXiv preprint arXiv:2312.00878*, 2023. 1, 2, 4, 6, 7, 8, 9, 10

[6] Yuri Y Boykov and M-P Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in ND images. In *Proceedings eighth IEEE international conference on computer vision. ICCV 2001*, pages 105–112. IEEE, 2001. 3, 6, 10

[7] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020. 2

[8] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9650–9660, 2021. 1, 2, 7

[9] Hila Chefer, Yuval Alaluf, Yael Vinker, Lior Wolf, and Daniel Cohen-Or. Attend-and-excite: Attention-based semantic guidance for text-to-image diffusion models. *ACM Transactions on Graphics (TOG)*, 42(4):1–10, 2023. 2, 4

[10] Minghao Chen, Iro Laina, and Andrea Vedaldi. Training-Free Layout Control With Cross-Attention Guidance. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 5343–5353, 2024. 4, 8

[11] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3213–3223, 2016. 2, 6, 10

[12] Inderjit S Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 269–274, 2001. 2, 6

[13] 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 *International Conference on Learning Representations*, 2020. 2, 3

[14] Mark Everingham and John Winn. The pascal visual object classes challenge 2012 (voc2012) development kit. *Pattern Anal. Stat. Model. Comput. Learn., Tech. Rep*, 2007:1–45, 2012. 2, 6

[15] Songwei Ge, Taesung Park, Jun-Yan Zhu, and Jia-Bin Huang. Expressive Text-to-Image Generation with Rich Text. In *IEEE International Conference on Computer Vision (ICCV)*, 2023. 2, 6

[16] Mark Hamilton, Zhoutong Zhang, Bharath Hariharan, Noah Snavely, and William T Freeman. Unsupervised semantic segmentation by distilling feature correspondences. In *International Conference on Learning Representations*, 2021. 2, 8

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 2, 3

[18] Wenbin He, Suphanut Jamonnak, Liang Gou, and Liu Ren. CLIP-S4: Language-Guided Self-Supervised Semantic Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11207–11216, 2023. 2

[19] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-or. Prompt-to-Prompt Image Editing with Cross-Attention Control. In *International Conference on Learning Representations*, 2023. 2, 4, 5

[20] Laurynas Karazija, Iro Laina, Andrea Vedaldi, and Christian Rupprecht. Diffusion Models for Zero-Shot Open-Vocabulary Segmentation. *arXiv preprint arXiv:2306.09316*, 2023. 2, 6, 7, 8, 9, 10

[21] Aliasghar Khani, Saeid Asgari Taghanaki, Aditya Sanghi, Ali Mahdavi Amiri, and Ghassan Hamarneh. SLiMe: Segment Like Me. *arXiv preprint arXiv:2309.03179*, 2023. 1, 2, 5

[22] Junho Kim, Byung-Kwan Lee, and Yong Man Ro. Causal unsupervised semantic segmentation. *arXiv preprint arXiv:2310.07379*, 2023. 7, 8

[23] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollar, and Ross Girshick. Segment Anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4015–4026, 2023. 1, 2, 11

[24] Neehar Kondapaneni, Markus Marks, Manuel Knott, Rogério Guimarães, and Pietro Perona. Text-image alignment for diffusion-based perception. *arXiv preprint arXiv:2310.00031*, 2023. 2, 4, 5

[25] Harold W Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, 1955. 5, 10

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

[27] Kehan Li, Zhennan Wang, Zesen Cheng, Runyi Yu, Yian Zhao, Guoli Song, Chang Liu, Li Yuan, and Jie Chen. ACSeg: Adaptive Conceptualization for Unsupervised Semantic Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7162–7172, 2023. 2, 6, 7, 8

[28] Yi Li, Hualiang Wang, Yiqun Duan, and Xiaomeng Li. Clip surgery for better explainability with enhancement in open-vocabulary tasks. *arXiv preprint arXiv:2304.05653*, 2023. 2, 4, 6, 7, 8, 9, 10

[29] 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 *European Conference on Computer Vision*, pages 740–755. Springer, 2014. 2, 6

[30] Luke Melas-Kyriazi, Christian Rupprecht, Iro Laina, and Andrea Vedaldi. Deep spectral methods: A surprisingly strong baseline for unsupervised semantic segmentation and localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8364–8375, 2022. 2, 6, 10

[31] 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 891–898, 2014. 2, 6

[32] Mark EJ Newman and Michelle Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69(2):026113, 2004. 2

[33] Quang Ho Nguyen, Truong Vu, Anh Tran, and Khoi Nguyen. Dataset Diffusion: Diffusion-based Synthetic Dataset Generation for Pixel-Level Semantic Segmentation. In *Advances in Neural Information Processing Systems*, 2023. 2, 5

[34] Or Patashnik, Daniel Garibi, Idan Azuri, Hadar Averbuch-Elor, and Daniel Cohen-Or. Localizing Object-level Shape Variations with Text-to-Image Diffusion Models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023. 4, 6, 10

[35] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 1, 2, 3, 4, 10

[36] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-Resolution Image Synthesis With Latent Diffusion Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022. 1, 2, 3, 4, 8, 10

[37] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015. 3

[38] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. "GrabCut" interactive foreground extraction using iterated graph cuts. *ACM transactions on graphics (TOG)*, 23(3): 309–314, 2004. 3, 6, 10

[39] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Raphael Gontijo-Lopes, Burcu Karagol Ayan, Tim Salimans, Jonathan Ho, David J. Fleet, and Mohammad Norouzi. Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding. In *Advances in Neural Information Processing Systems*, 2022. 11

[40] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905, 2000. 2, 6, 10

[41] Gyungin Shin, Weidi Xie, and Samuel Albanie. Namedmask: Distilling segmenters from complementary foundation models. *arXiv:2209.11228*, 2022. 2, 6, 7, 8

[42] Gyungin Shin, Weidi Xie, and Samuel Albanie. Reco: Retrieve and co-segment for zero-shot transfer. *arXiv preprint arXiv:2206.07045*, 2022. 1, 2, 6, 7, 8, 9

[43] Oriane Siméoni, Gilles Puy, Huy V Vo, Simon Roburin, Spyros Gidaris, Andrei Bursuc, Patrick Pérez, Renaud Marlet, and Jean Ponce. Localizing objects with self-supervised transformers and no labels. *arXiv preprint arXiv:2109.14279*, 2021. 2

[44] Meng Tang, Abdelaziz Djelouah, Federico Perazzi, Yuri Boykov, and Christopher Schroers. Normalized cut loss for weakly-supervised cnn segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1818–1827, 2018. 6

[45] Raphael Tang, Linqing Liu, Akshat Pandey, Zhiying Jiang, Gefei Yang, Karun Kumar, Pontus Stenetorp, Jimmy Lin, and Ferhan Ture. What the DAAM: Interpreting stable diffusion using cross attention. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2023. 4

[46] Junjiao Tian, Lavisha Aggarwal, Andrea Colaco, Zsolt Kira, and Mar Gonzalez-Franco. Diffuse, Attend, and Segment:

Unsupervised Zero-Shot Segmentation using Stable Diffusion. *arXiv preprint arXiv:2308.12469*, 2023. 1, 2, 4, 5, 6, 7, 8, 9, 10

[47] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention, 2021. 7

[48] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017. 1, 2, 3

[49] Jinglong Wang, Xiawei Li, Jing Zhang, Qingyuan Xu, Qin Zhou, Qian Yu, Lu Sheng, and Dong Xu. Diffusion model is secretly a training-free open vocabulary semantic segmenter. *arXiv preprint arXiv:2309.02773*, 2023. 1, 2, 5, 6, 7, 8, 10

[50] Xudong Wang, Rohit Girdhar, Stella X Yu, and Ishan Misra. Cut and learn for unsupervised object detection and instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3124–3134, 2023. 6

[51] Size Wu, Wenwei Zhang, Lumin Xu, Sheng Jin, Xiangtai Li, Wentao Liu, and Chen Change Loy. CLIPSelf: Vision Transformer Distills Itself for Open-Vocabulary Dense Prediction. *arXiv preprint arXiv:2310.01403*, 2023. 9, 11

[52] 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. *arXiv preprint arXiv:2303.11681*, 2023. 2

[53] Changming Xiao, Qi Yang, Feng Zhou, and Changshui Zhang. From Text to Mask: Localizing Entities Using the Attention of Text-to-Image Diffusion Models. *arXiv preprint arXiv:2309.04109*, 2023. 5

[54] Hu Xu, Saining Xie, Xiaoqing Ellen Tan, Po-Yao Huang, Russell Howes, Vasu Sharma, Shang-Wen Li, Gargi Ghosh, Luke Zettlemoyer, and Christoph Feichtenhofer. Demystifying clip data. *arXiv preprint arXiv:2309.16671*, 2023. 8, 10

[55] Jiarui Xu, Sifei Liu, Arash Vahdat, Wonmin Byeon, Xiaolong Wang, and Shalini De Mello. Open-Vocabulary Panoptic Segmentation With Text-to-Image Diffusion Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2955–2966, 2023. 2

[56] Chenglin Yang, Siyuan Qiao, Yuan Cao, Yu Zhang, Tao Zhu, Alan Yuille, and Jiahui Yu. IG Captioner: Information Gain Captioners are Strong Zero-shot Classifiers. *arXiv preprint arXiv:2311.17072*, 2023. 11

[57] Ryota Yoshihashi, Yuya Otsuka, Tomohiro Tanaka, et al. Attention as annotation: Generating images and pseudo-masks for weakly supervised semantic segmentation with diffusion. *arXiv preprint arXiv:2309.01369*, 2023. 8

[58] Chong Zhou, Chen Change Loy, and Bo Dai. Extract free dense labels from clip. In *European Conference on Computer Vision (ECCV)*, 2022. 1, 2, 4, 5, 6, 7, 8, 9, 10